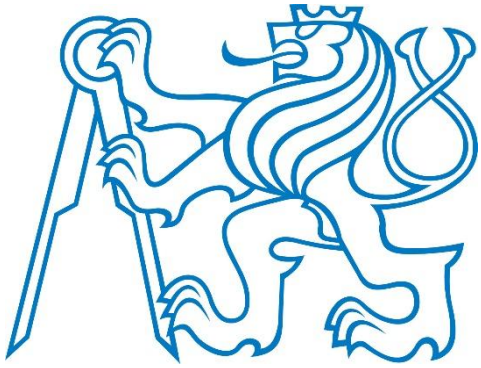
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ



Diplomová práce

Simulace robotické ruky pro automatizaci testování senzoru gest

Robotic arm simulation for gesture sensor automatic testing

Autor: Bc. Lukáš Poláček

Vedoucí diplomové práce: Ing. Martin Nečas, MSc. Ph.D.

Akademický rok: 2016/2017

PROHLÁŠENÍ

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod pro užití tohoto díla ve smyslu § 60 zákona č. 121/2000SB., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne …… ……

podpis

PODĚKOVÁNÍ

Děkuji svým blízkým, rodině a přátelům za jejich trpělivost a pochopení. Především děkuji svým rodičům za neustálou podporu.

Děkuji také svému vedoucímu práce, Ing. Martinu Nečasovi, MSc. Ph.D., za rady a čas, které mi věnoval. Jeho klid mi dodával důvěru v dokončení.

ANOTACE

Diplomová práce se zabývá simulací pohybů ruky pro testování senzoru gest v nové řadě automobilu Volkswagen Golf R Touch.

V první části je zpracována rešerše humanoidních rukou a robotických ramen, která by byla vhodná pro hardwarové řešení automatizace testování senzoru gest.

V druhé části je popsána simulace provedená v softwaru Matlab Simulink s využitím toolboxu Simscape přesněji řečeno SimMechanics. Zde je i ukázka gest, která se používají k ovládání infotainmentu vozu.

V následné části je projekt rozebrán z hlediska softwarového inženýrství. Zde se odhaduje cena a trvání projektu.

V poslední části je rozebírána možnost řešení inverzní kinematiky pomocí optimalizačních metod. Je zde proveden výpočet pomocí metod pro globální a lokální optimalizaci. Pro globální optimalizaci je použit genetický algoritmus a pro lokální optimalizaci simplexová metoda.

ABSTRACT

The thesis deals with gesture simulation for gesture sensor testing in the new model of the Volkswagen Golf R Touch.

In the first part, a research of humanoid hands and robotic arms is processed. These robots would be suitable for the hardware solution of gesture sensor automation.

The second part describes simulations made in Matlab Simulink software using Simscape toolbox, more precisely SimMechanics. Here is a sample of gestures that are used to control the vehicle's infotainment.

In the following part, the project is described from the point of view of software engineering. Here, the cost and duration of the project are estimated.

The last part analyzes the possibility of inverse kinematics by means of optimization methods. Calculation is done using global and local optimization methods. Genetic algorithm is used for global optimization, and the simplex method for local optimization.

OBSAH

Obsah

Prohlášení 2
Poděkování 3
Anotace 4
Abstract 5
Obsah 6
1 Úvod 1
2 Rešerše pracovních hlavic a robotických ramen 2
2.1 Pracovní hlavice 2
2.1.1 Pasivní úchopné hlavice 3
2.1.2 Aktivní úchopné hlavice 3
2.2 Humanoidní ruka 7
2.2.1 ADA 8
2.2.2 Brunel hand 9
2.2.3 Mechax 10
2.2.4 Biogrip 11
2.2.5 AR10 12
2.2.6 SVH robotic hand 13
2.3 Robotické rameno 14
2.3.1 IRB 120 16
2.3.2 UR3 17
2.3.3 KR 6 R700 FIVVE 18
2.3.4 EVA 19

3 Testování senzoru gest 20
3.1 Řešení zadaného úkolu 21
3.2 Model 21
3.3 Mechanický model 22
3.4 Funkční model v prostředí Matlab Simulink 23
3.4.1 Vytvoření modelu v SimMechanics 23
3.5 Výsledná gesta 26
3.5.1 Iniciační poloha 26
3.5.2 Startovací gesto 26
3.5.3 One finger gesto 27
3.5.4 Two finger gesto 27
3.5.5 Wave gesto 28
3.5.6 Change song gesto 28
4 Softwarové inženýrství 29
4.1 Deklarace záměru projektu 29
4.2 Odborný článek 30
4.2.1 Uživatelské role 30
4.3 Datový slovník 32
4.3.1 Gesture 32
4.3.2 Model 32
4.3.3 Simulation 33
4.3.4 Car infotainment 33
4.4 Řešitelský tým 35
4.5 Návrh řešení a rozpočet 35
4.5.1 COCOMO 35

4.5.2 Výsledná cena 35
4.6 Časový harmonogram 37
5 Výpočet inverzní kinematiky 41
5.1 Maticová metoda 41
5.1.1 Maticová metoda ve 2D 41
5.1.2 Maticová metoda ve 3D 42
5.1.3 Použití více tranformací 43
5.1.4 Souřadnice 44
5.2 Optimalizace 45
5.2.1 Cílová funkce 45
5.3 Metody lokální a globální optimalizace 46
5.3.2 Genetický algoritmus 48
5.3.3 Simplexová metoda 51
6 Závěr 56
7 Zdroje 57
8 Přílohy 58
8.1 Model v simulinku 58
8.1.1 Model top 58
8.1.2 Model bottom 59
8.1.3 Ovladani1 60
8.1.4 Sphere1 60
8.1.5 Ovládání2 61
8.1.6 Předloktí s dlaní 62
8.1.7 Sphere 2 62
8.1.8 Ovládání3 63

8.1.9 Sphere3 63
8.1.10 Ovládání palec 64
8.1.11 Palec 64
8.1.12 Ovládání prst 1 65
8.1.13 Prst1 65
8.2 Matlab skripty 66
8.2.1 konstants 66
8.2.2 maticovy_zapis 66
8.2.3 myfitness 69
8.2.4 genetic algorithm + fminsearch 70
8.2.5 LB 70
8.2.6 UB 71
8.2.7 fmincon 72
8.2.8 init 72
8.2.9 grip 74
8.2.10 change song 75
8.2.11 onefingerpoint 77
8.2.12 twofingerpoint 79
8.2.13 start 79
8.2.14 wave 80
8.2.15 thumb up 84

1 ÚVOD

V této práci se píše o simulaci gest pro testování senzoru gest v novém voze Volkswagen Golf R touch. Pomocí tohoto senzoru se dá ovládat infotainment vozu. Jako například stahování okének, změna podbarvení palubní desky, ovládání hlasitosti, změna skladby.

Takto ovládaný systém infotainmentu se začíná objevovat i u dalších automobilek a jeho vývoj včetně testů je otázkou konkurenčního boje.

Z hlediska finančních a personálních zdrojů je automatizace těchto testů nutností. Manuální testování by bylo pro testera nepříjemné z hlediska nenáročnosti úkonů, které by musel předvádět a potřeby vykonávat tyto úkony repetetivně v dlouhých časových úsecích.

V další části práce je vypracovaná analýza z hlediska softwarového inženýrství, kde jsou zpracovány informace o časové náročnosti projektu, přibližných finančních požadavcích na klienta a personálních požadavcích ve firmě.

V poslední části se rozebírá možnost výpočtu inverzní kinematiky pomocí optimalizačních metod a následný výpočet pomocí metody pro nalezení globálních extrémů metodou genetických algoritmů. Následně je aplikována metoda pro hledání lokálních extrémů funkcí fminsearch na již vypočítaných odhadech z optimalizace pomocí genetických algoritmů. Tento výpočet je srovnán s použitím funkce fmincon, která respektuje okrajové podmínky na rozdíl od funkce fminsearch.

2 REŠERŠE PRACOVNÍCH HLAVIC A ROBOTICKÝCH RAMEN

2.1 PRACOVNÍ HLAVICE

Pracovní hlavice představují endefektor pohybového systému průmyslového robota a manipulátoru. Můžeme je dělit dle aplikace a dle typu operací.

Podle aplikace je možné je rozdělit na hlavice pro vkládání objektu, mezioperační manipulace, technologické operace a kontrolní operace. Aplikace můžeme provádět sériově nebo paralelně, tedy současně, což nám dává úsporu času.

Dle typu operací je můžeme dělit na úchopné pracovní hlavice, technologické hlavice, které drží nástroj, poté kontrolní a měřící hlavice, kombinované hlavice, které mají více funkcí a speciální hlavice.

Z hlediska použitelnosti je možné dělení na jednoúčelové, tyto hlavice jsou atypické pro daný objekt, víceúčelové, používají se pro založení polotovaru, vyjmutí výrobku atd., univerzální, tyto se dají přednastavit nebo přizpůsobit, a antropomorfní úchopné hlavice.

Na úchopné hlavici se nacházejí úchopné prvky. Tyto prvky dělíme dle styku a dle vyvození úchopné síly. Dle styku je můžeme dělit na mechanické, magnetické a podtlakové. Mechanické se chovají jako lidská ruka. Magnetické potřebují feromagnetický materiál. Podtlakové vyžadují relativně hladký povrch. Také je můžeme dělit podle počtu úchopných prvků na dvou, tří a více prvkové, podle počtu kontaktních rovin a charakteru uchopení. Charakterů uchopení máme několik druhů tvarové, silové (tření mezi efektorem a předmětem), kombinované, to se používá nejčastěji, a dle trajektorie chodu prvků na lineární (paralelní úchop), radiální a úhlový úchop.

Rozdělení dle vyvození úchopné síly je na pasivní, k čemu se používá například guma nebo přísavka, a na aktivní neboli s ovládáním, k čemu se používají například mechanické čelisti.

2.1.1 PASIVNÍ ÚCHOPNÉ HLAVICE

Pasivní úchopné hlavice je možné zkonstruovat s pevnými a stavitelnými opěrami, pružnými a odpruženými čelistmi, permanentními magnety nebo deformačními přísavkami.

Mechanické pasivní úchopné hlavice jsou nejjednodušší. Reprezentantem může být hák nebo například na ruce je pasivním prvkem dlaň.

Magnetické pasivní úchopné hlavice mají vsazené permanentní tyčové magnety, které lze dělit podle tvaru. Používají se převážně k manipulaci s lehčími feromagnetickými předměty. Problém představuje ulpívání drobných částic na magnetu.

Podtlakové pasivní úchopné hlavice obsahují pryžové deformační přísavky. Používají se na manipulaci s rovnými deskami, plechy, skly a krabicemi. Pracovní prostředí může být plynné nebo kapalné.

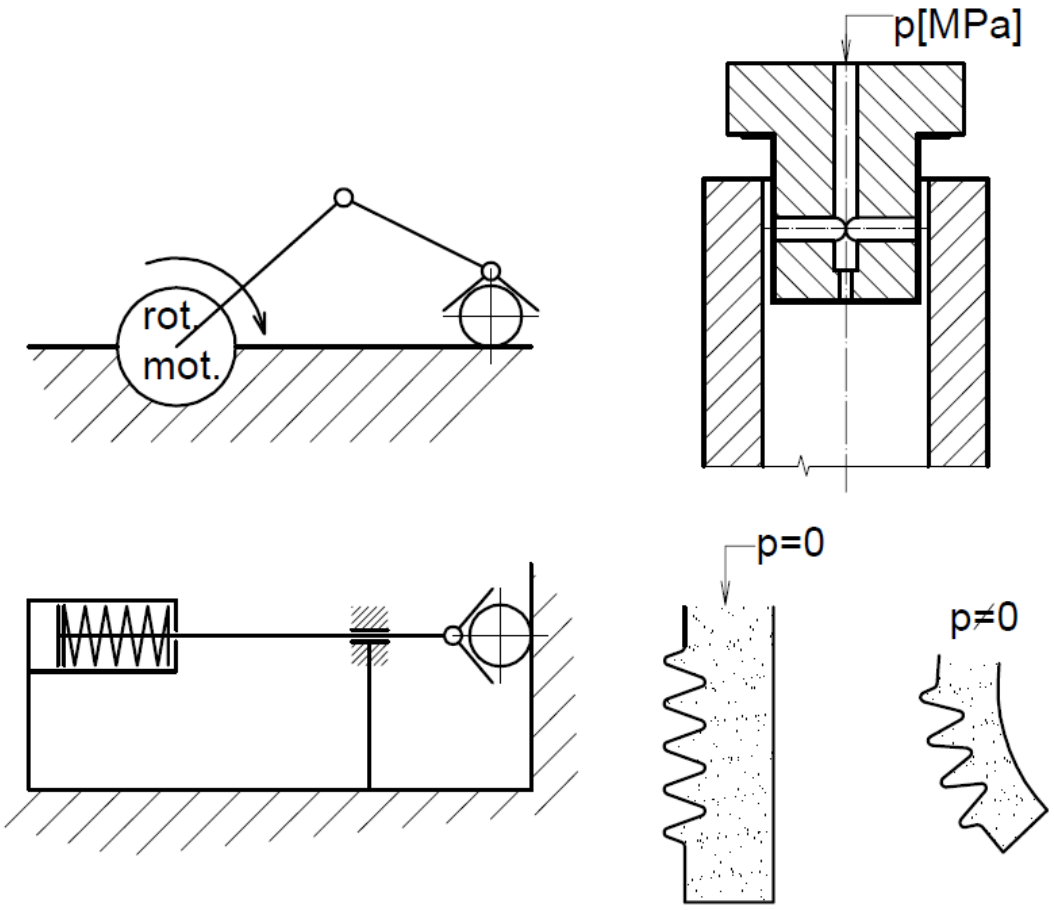
2.1.2 AKTIVNÍ ÚCHOPNÉ HLAVICE

Aktivní úchopné hlavice většinou obsahují motor transformační blok a úchopný prvek.

Pod pojmem „transformační blok“ rozumíme mechanický převod.

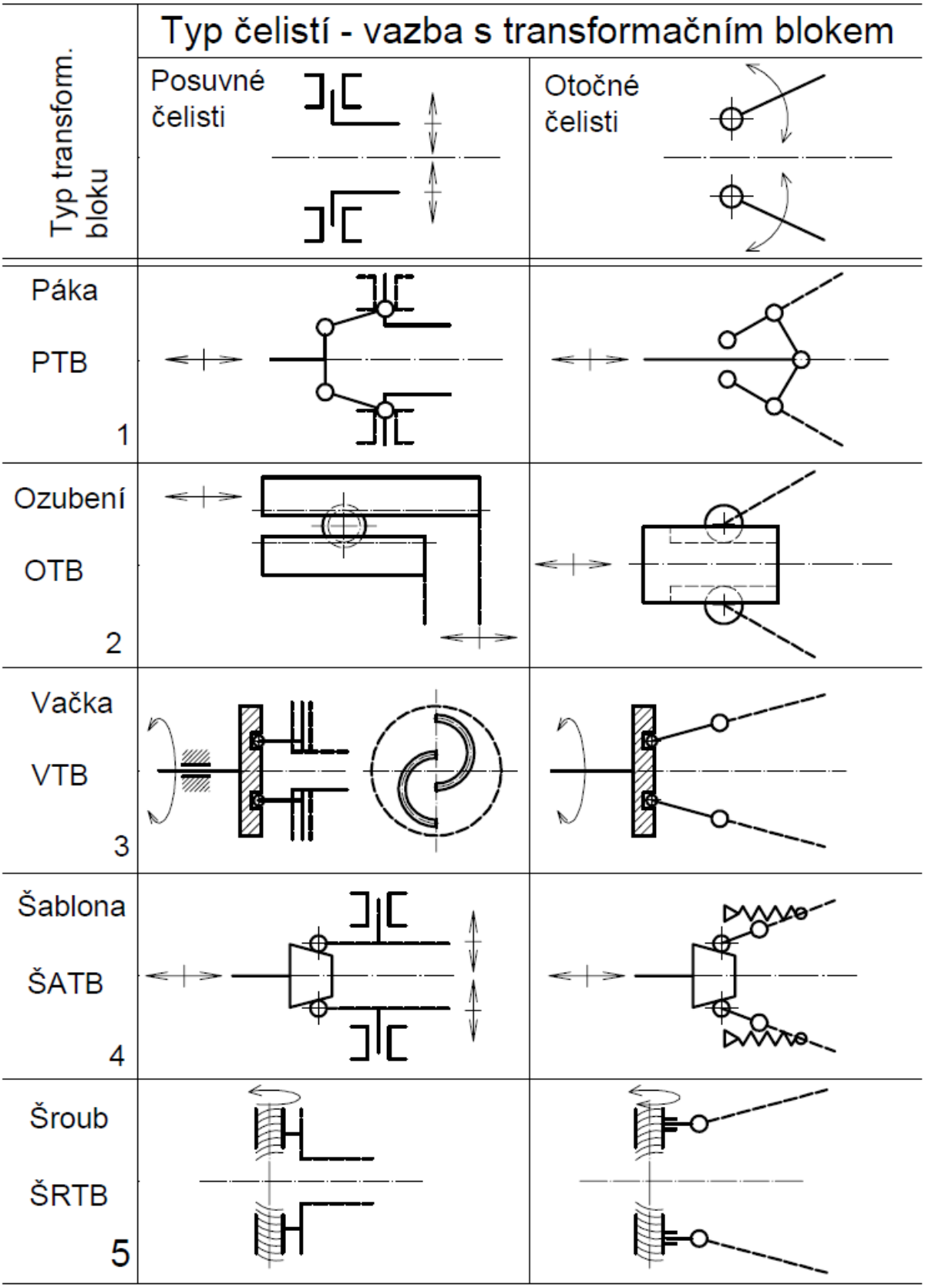
Úchopný prvek může být čelist, elektromagnet nebo podtlaková komora. Čelisti dělíme dle druhu jejich pohonu na hydraulické, pneumatické, elektromagnetické a kombinované (viz obr. 1).

Mohou se vyskytovat i aktivní úchopné hlavice bez transformačního bloku. Ty mají většinou jen jeden aktivní prvek. Možná konstrukce je oddělené uspořádání nebo integrace motoru do úchopného prvku.



Obrázek 1: Aktivní úchopné prvky[1]

Transformační blok můžeme zkonstruovat jako pákový, ozubený, vačkový, šablonový a šroubový (viz obr. 2). Může mít konstantní nebo variabilní převod. Ve vazbě s motorem může umožňovat změnu druhu pohybu, změnu smyslu pohybu, změnu úchopné síly, změnu rychlosti pohybu. A společné ovládání více úchopných prvků.



Obrázek 2: Typy čelistí s transformačním blokem[1]

Nejčastěji se používají dvouprsté grippery (příklad viz obr. 3). Jedná se o jednoduchou konstrukci, která dokáže manipulovat s malými předměty o váze pár gramů až desítek kilogramů.

Obrázek 3: dvouprstý gripper[6]

Dále velice používaným typem gripperu jsou tříprsté hlavice (příklad viz obr. 4). Ty mají výhodu v tom, že mají o jednu třecí plochu navíc a automaticky dochází k vystředění rotačních součástí. To se s výhodou používá pro přesnější zajištění polohy manipulovatelného objektu.

Obrázek 4: tříprstý gripper[7]

2.2 HUMANOIDNÍ RUKA

Lidská ruka je bezpochyby jedním z nejuniverzálnějších a nejsložitějších nástrojů v přírodě. Vytvoření nástroje obdobných vlastností je velkou výzvou pro generace výzkumných pracovníků a konstruktérů a výzkum a vývoj robotických rukou je jedním z nejsledovanějších problémů v humanoidní robotice. V poslední době došlo v této oblasti vývoje k významnému pokroku. Realitou jsou robotické ruce, které napodobují jak anatomii, tak pohyby a funkce lidské ruky včetně jemného úchopu. Nejnovější výsledky vývoje okazují, že víceprsté ruce budou brzy součástí běžného pracovního prostředí a v současnosti používané jednoduché robustní dvou či tříprsté koncové efektory budou nahrazeny pětiprstýma rukama, schopnýma provádět složité, přesné a jemné činnosti. Výroba víceprstých rukou se zvlášť řízenými prsty a klouby, jak je tomu u lidské ruky, je umožněna pokrokem v mikromechanice a mikroelektronice. Vývoj v dílčích oblastech mechatroniky, která je založená na funkční a prostorové integraci mechaniky, elektroniky a řízení, nyní již dospěl tak daleko, že robotická ruka již může být z velké části zkonstruována ze standardních komerčních součástí. [3]

Pro testování senzoru gest se vybírali různé robotické ruce. Jelikož bylo potřeba testovat gesta lidské ruky, musely být vybírány robotické ruce humanoidního typu.

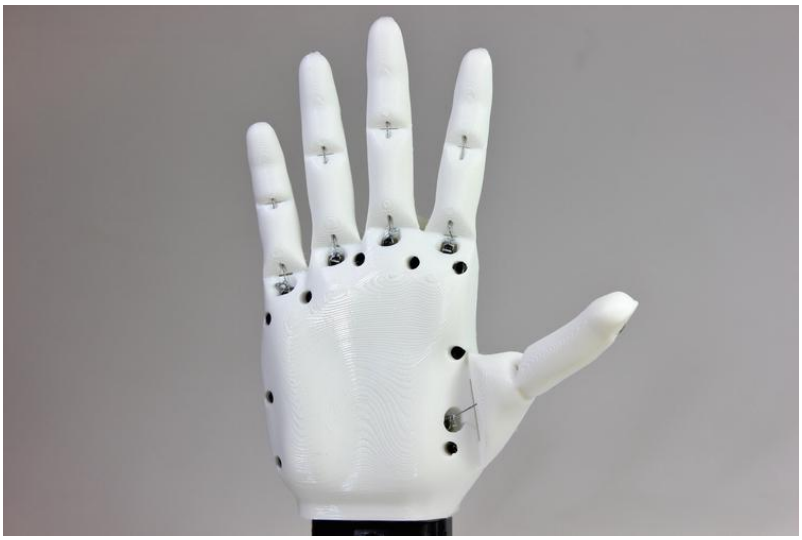
2.2.1 ADA

Robotická ruka ADA je od firmy Open Bionics se sídlem v Bristolu. Jedná se o výukového robota pro protetiku. Další možnost využití výrobce udává pro humanoidního robota nebo pro interakci mezi robotem a člověkem.

Výroba 3D tiskem je jednou z nejlevnějších metod výroby.

Proto je tento robot vhodný pro náš projekt. Je však nutné zvážit, zda nepohyblivé poslední články nebudou překážkou, která by mohla vyřadit ruku z výběru.

Díky své váze pouhých 380g je výborným kandidátem na levnou variantu řešení tohoto projektu.



Obrázek 5: Robotická ruka ADA[8]

+ 5 stupňů volnosti
+ 5 motorů
+ Cena £569 neboli 18 000 Kč
+ Smontovatelnost v řádu minut s použitím standardních nástrojů
+ Konektivita přes USB
+ Kompatibilita s Arduino

– Nepohyblivé poslední články

2.2.2 BRUNEL HAND

Dalším výrobkem z firmy Open Bionics je robotická ruka Brunel hand. Tato ruka je také tištěna na 3D tiskárně s dodávanými komponenty, jako jsou aktuátory, desky plošných spojů a dalšího příslušenství.

Obrázek 6: Robotická ruka Brunel[8]

+ 9 stupňů volnosti
+ Mohutnější palec
+ Plně open source hardware a software
+ Kompatibilita s Arduino IDE
+ Programovatelné přes USB
+ Váha 371g
+ Proudová zpětná vazba na všech motorových kanálech
+ Robustní a mrazuvzdorná konstrukce
+ Protiskluzový povrch úchopných ploch

– Pouze 4 aktuované stupně
– Cena £1499 neboli 47 400 Kč

2.2.3 MECHAX

Jedná se o robotickou ruku, která je prodávaná jako hračka na stránkách Roboshop.com. Tato ruka je vyrobena z kovu. Jedná se o demonstrátor pohybu ruky.

Obrázek 7: robotická ruka mechax[9]

+ 5 stupňů volnosti
+ 5 motorků
+ Mezní rychlost každého prstu v plném rozsahu je přibližně 1 Hz
+ Možnost 3D tisku částí
+ Motorky, které jsou doporučené pro aplikace s držením
+ Všechny části vyrobeny na CNC z leteckého hliníku, černěny

– Cena 950 $ neboli 24000 Kč
– Možné špatné odečítání polohy prstů z důvodu malé plochy prstu

2.2.4 BIOGRIP

Jedná se o robotickou ruku od stejného poskytovatele Roboshop.com. Na rozdíl od předchozí má prsty zkonstruované tak, aby byli podobné lidským.

Obrázek 8: Robotická ruka Biogrip[10]

+ 5 stupňů volnosti
+ 5 aktuátorů
+ Palec má 2 stupně volnosti oba aktuované
+ Mechanický systém umožňuje imitovat všechny lidské pohyby
+ Palec může změnit pozici, aby se adaptoval na různé úchytové strategie
+ Umožňuje jednoduchou montáž a redukuje údržbu
+ Nezávislé prsty
+ Možnost naistalování senzorů do prstů
+ Váha 550g

– Pouze 3 prsty
– Cena 1550 $ neboli 40000 Kč

2.2.5 AR10

Jedná se o robotickou ruku prodávanou na stránkách Active-robots.com, kde je prodávána jako edukační pomůcka.

Obrázek 9: Robotická ruka AR10[11]

+ 10 stupňů volnosti
+ 10 aktuátorů
+ Konektivita přes USB
+ Konstrukce z anodizovaného hliníku a plastu

– Cena £3600 neboli 114 000Kč
– Možné špatné odečítání senzoru kvůli neplným článkům prstů

2.2.6 SVH ROBOTIC HAND

Jedná se o robotickou ruku vyráběnou firmou Schunk. Co se robotické ruky týče tak na trhu nebyl nalezen lepší výrobek pro testování senzoru gest. S váhou 1,3 kg se řadí k nejtěžším z robotických rukou v této práci porovnávaných.

Obrázek 10: Robotická ruka SVH[12]

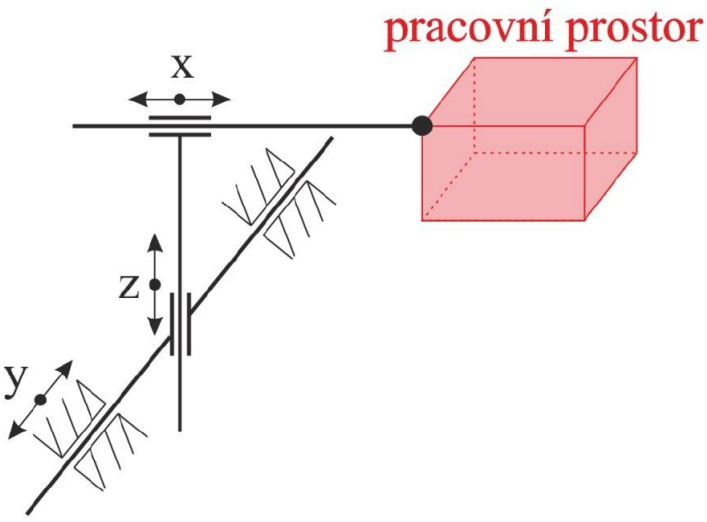
+ 20 kloubů
+ 9 motorků
+ 21 stupňů volnosti
+ Elastické uchopovací povrchy
+ Normalizované propojení na úchyt robotického ramena

– Cena 1 230 300 Kč

2.3 ROBOTICKÉ RAMENO

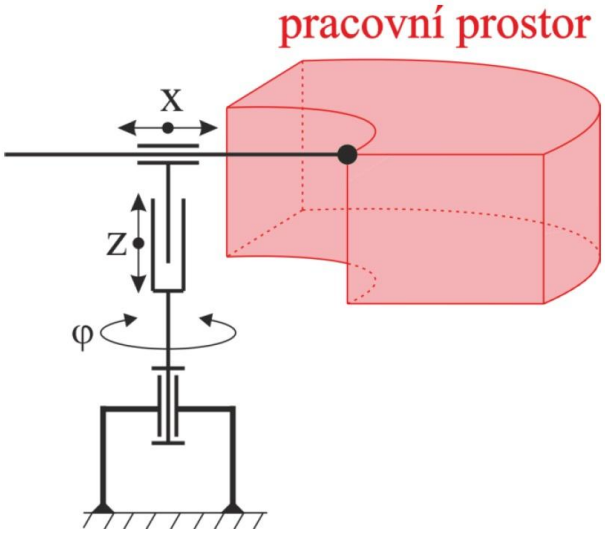
Pro manipulaci s břemeny se v průmyslu už dlouho používají různé robotické systémy. Rozdělují se podle pohybů, které jsou schopny vykonávat.

První skupinou jsou roboty tyku K (kartesián) se třemi navzájem kolmými translacemi. Pracovní prostor tohoto typu tvoří kvádr (viz obr. 11).



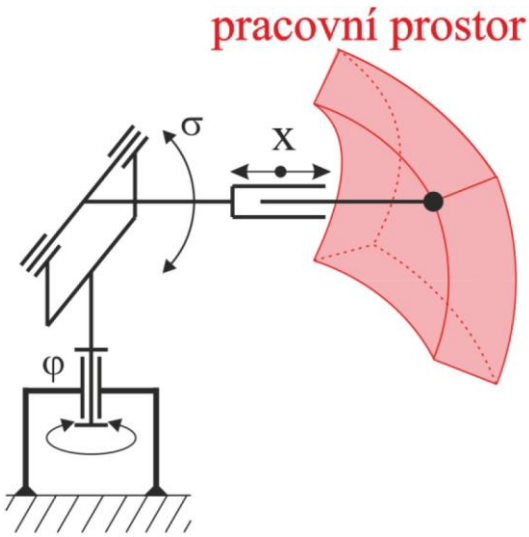
Obrázek 11: Typ K[1]

Druhou skupinou jsou roboty typu C (cylindric) s dvěma translacemi a jednou rotací. Výsledný pracovní prostor je válec (viz obr. 12).



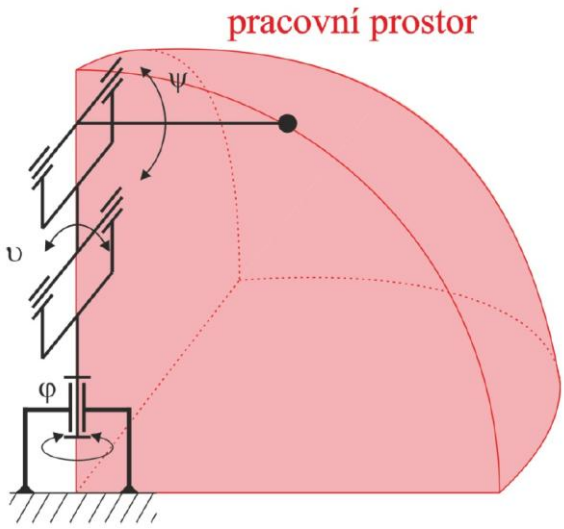
Obrázek 12: Typ C[1]

Třetí skupinou jsou roboty typu S (sferical) se dvěma rotacemi a jednou translací. Výsledný pracovní prostor je koule (viz obr. 13).



Obrázek 13: Typ S[1]

Poslední skupinou jsou roboty typu A (angular) se třemi rotacemi (viz obr. 14).



Obrázek 14: Typ A[1]

Tento typ se hodí pro naši aplikaci avšak je potřeba zvýšit počet stupňů volnosti robota, proto vybíráme roboty s 5 a více stupni volnosti.

2.3.1 IRB 120

Prvním představovaným robotickým ramenem je IRB 120 od výrobce ABB. Tento robot váží 25 kg, má dosah 580 mm a maximální zatížení jsou 3 kg.

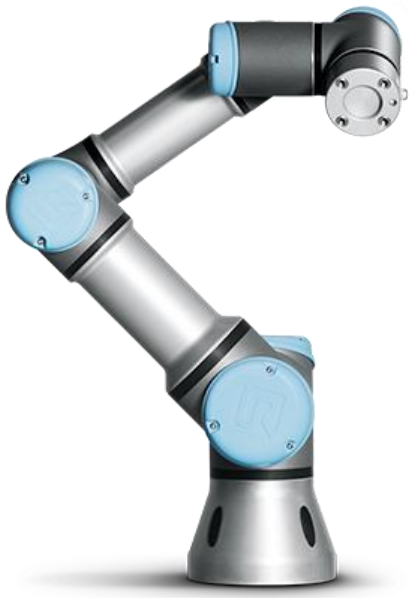
Obrázek 15: IRB 120[13]

+ Vysoká rychlost pohybu
+ Opakovatelnost na 10 mikronů
+ 6 stupňů volnosti

– Cena 435 000 Kč

2.3.2 UR3

Druhým robotickým ramenem připadajícím v úvahu je UR3 od společnosti Universal robots. Tento robot váží 11 kg s dosahem 500 mm a maximálním zatížením 3 kg.



Obrázek 16: UR3[14]

+ Kolaborativní robot, což je robot, který nemusí být v kleci, protože při kontaktu se sám zastaví, tudíž se snižuje nebezpečí pro obsluhu.
+ 6 stupňů volnosti

– Cena 405 375 Kč

2.3.3 KR 6 R700 FIVVE

Dalším robotickým ramenem v této práci zmiňovaným je KR 6 R700 FIVVE od společnosti Kuka. Tento robot váží 48 kg a má dosah 706,7 mm s maximálním zatížením 6 kg.

Obrázek 17: KR 6 R700 FIVVE[15]

+ Firma vlastní již 2 Kuka roboty
– Pouze 5 stupňů volnosti

2.3.4 EVA

Posledním robotickým ramenem, které je zde zmiňováno, je EVA od společnosti Automata. Tento robot je zajímavý tím, že většina jeho součástí je tištěna na 3D tiskárně, což velmi zlevňuje výrobu. Tento robot také váží pouze 2,5 kg s dosahem 706,7 mm a maximálním zatížení 0,75 kg.

Obrázek 18: EVA[16]

+ Cena 3000$ neboli 75 240 Kč
+ Konektivita přes USB nebo Wi-fi
+ Opensource řídící systém
+ 6 stupňů volnosti

– Čas dodání, jedná se o start up
– Malé maximální zatížení

3 TESTOVÁNÍ SENZORU GEST

Motivací bylo vytvoření systému pro automatické testování senzoru gest umístěném ve vozidle Volkwagen Golf R touch. Tento systém by měl simulovat pohyby lidské ruky, kterou řidič vozu ovládá infotainment. Senzor je monochromatická kamera, která je umístěna u řadicí páky. Snímá prostor nad řadicí pákou, tedy v prostoru, který je řidiči dobře přístupný a pohodlně v něm dokáže vytvářet gesta, která kamera zachytí. Zachycený obraz pak prochází softwarem na rozpoznávání obrazu a ten následně vydává příkazy do infotainmentu.

Zadaná gesta pro vytvoření testování byla ovládání pravého a levého okénka, změna podbarvení palubní desky, tato gesta jsou si podobná jen se provádí v jiném menu. Další jsou změna hlasitosti rádia a přepínání mezi skladbami. V neposlední řadě zahajovací gesto, po kterém software rozpozná, že má začít vydávat příkazy infotainmentu.

Ovládání levého okénka a změna podbarvení palubní desky se provádí pomocí gesta, kdy je zdvižen ukazováček, ostatní prsty jsou zaťaty v pěst a ruka se pohybuje zleva doprava a zpět.

Ovládání pravého okénka a druhé sady diod na podbarvení se provádí pomocí gesta, kdy jsou zdviženy dva prsty, konkrétně ukazováček a prostředníček. Ostatní prsty jsou zaťaty v pěst a ruka se pohybuje stejně jako v předchozím případě.

Změna hlasitosti se ovládá pomocí gesta, které je vytvořeno dlaní s nataženými prsty a pohybem ruky v lokti nahoru a dolů.

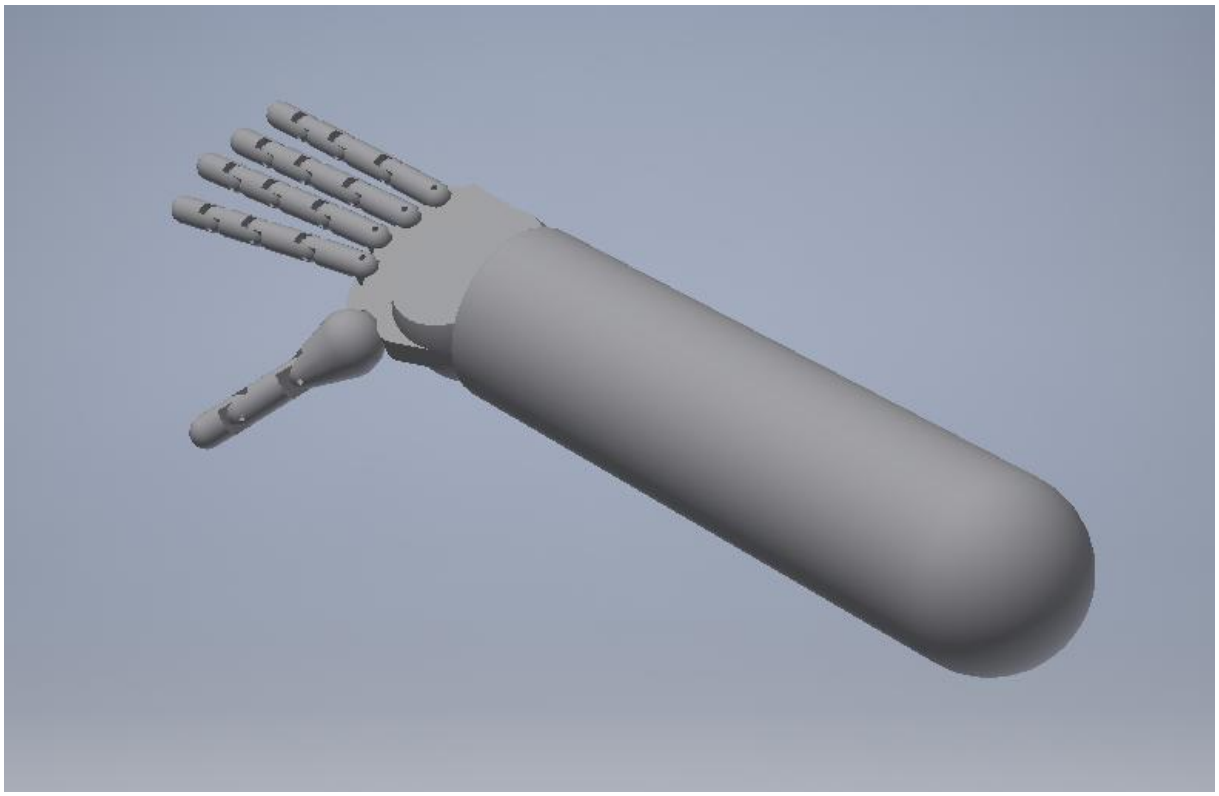
Přepínání mezi skladbami je umožněno gestem, kdy je dlaň otočena do vertikální pozice s nataženými prsty a celá ruka se pohybuje zleva doprava a zpět.

3.1 ŘEŠENÍ ZADANÉHO ÚKOLU

Daný úkol lze řešit pomocí robotického manipulátoru s gripperem ve tvaru humanoidní ruky. Tyto manipulátory a grippery jsou rozebrány v rešerši. Avšak řešení automatického testování pomocí hardwarového řešení a nákupu nového robota s gripperem ve tvaru humanoidní ruky nebylo z finančního hlediska možné, a tak bylo navrženo řešení pomocí simulace. To spočívá v promítání simulace pohybu modelu ruky, který je simulován v prostředí Matlab Simulink s použitím toolboxu Simscape převážně Simmechanics na monitor. Prostředí Simmechanics umožňuje importovat model vytvořený v konstrukčním programu Autodesk Inventor.

3.2 MODEL

Model vytvořený v prostředí Autodesk Inventor byl vytvořen tak, aby co nejvíce odpovídal vzhledem humanoidní ruce a co možná nejpřesněji napodoboval pohyby lidské ruky tak, aby senzor bez problémů rozpoznal zadaná gesta (viz obr. 19).



Obrázek 19: Model ruky v Autodesk Inventoru

3.3 MECHANICKÝ MODEL

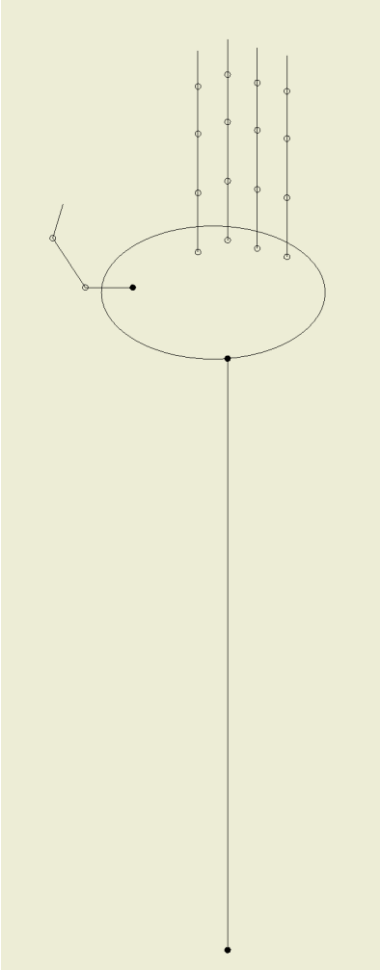
Mechanický model byl vytvořen tak, aby co nejvíce odpovídal mechanice lidské ruky a umožnil vytvořit simulaci, kterou by senzor rozpoznal, jako pohyby řidiče, a tím otestoval funkčnost právě daného senzoru. Byly použity tři kulové klouby (černé tečky viz obr. 20).

První kulový kloub se nachází v lokti, jelikož daný mechanismus začíná právě v lokti, je nutné mít možnost natáčet v tomto bodě ve všech třech osách.

Dále je kulový kloub umístěn v zápěstí, kde je potřeba využít všech os rotace pro daná gesta.

Poslední kulový kloub je umístěn v kořeni palce. Jeho využití je v gestech nezbytné pro simulaci všech pohybů lidského palce.

Zbylé klouby jsou vytvořeny jako rotační vazby (tečky bez výplně viz obr. 20).



Obrázek 20: Mechanické schéma ruky

Máme 21 těles se 3 sférickými a 18 rotačními vazbami.

$$6 \times n_{těles} - 3 \times n_{sfér} - 5 \times n_{rot} = n°_{volnosti} \quad (1)$$

Po výpočtu zjistíme, že zbývá 27 stupňů volnosti. Každý stupeň volnosti bude samostatně aktuovaný.

3.4 FUNKČNÍ MODEL V PROSTŘEDÍ MATLAB SIMULINK

Pro simulaci bylo zapotřebí přenést model z prostředí konstrukčního softwaru Autodesk Inventor do prostředí Matlab Simulink. Pro vytváření mechanických modelů v tomto programu je výhodné využít toolboxu Simscape. Z tohoto toolboxu se hodí nejvíce část SimMechanics.

3.4.1 VYTVOŘENÍ MODELU V SIMMECHANICS

Model vytvořený v prostředí Autodesk Inventor bylo nutné převést do prostředí toolboxu SimMechanics. K tomu bylo zapotřebí programu **smlink**, který bylo nutné nainstalovat do softwaru Autodesk Inventor i do Matlabu. Tento program vytvoří soubor slx, který byl dále nahrán do prostředí Matlabu. Po nahrání se vygeneruje model v programu Matlab Simulink, již tvořený součástmi z toolboxu SimMechanics.

3.4.1.1 PROPOJENÍ SIMULINKOVÝCH A FYZIKÁLNÍCH PROMĚNNÝCH

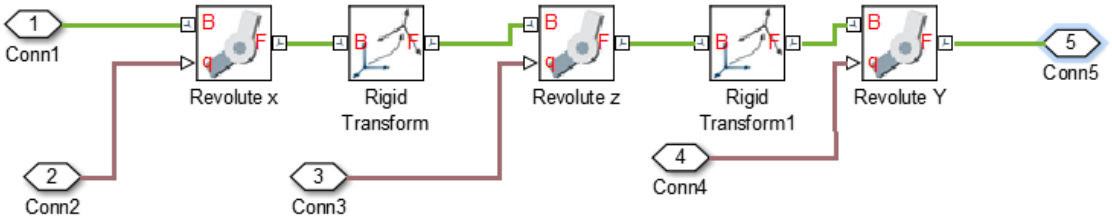
V prostředí Simscape se tvoří dvě sady odlišných proměnných. Simulinkové, se kterými se běžně v Simulinku pracuje, a fyzikální, se kterými pracují pouze bloky z toolboxu Simscape. K přechodu od jedné sady proměnných k druhé je zapotřebí speciálního blok S-PS nebo obráceného bloku PS-S (viz obr. 21). Díky těmto blokům je možné připojovat i jiné bloky, které jsou v prostředí Matlab Simulink běžné.

Obrázek 21: Blok pro konverzi Simulink to Physical

3.4.1.2 BLOKY SIMMECHANICS

V simulačním prostředí SimMechanics se nacházejí bloky s objekty, vazbami, transformace souřadnic, klouby, síly a momenty, převody a utility. V tomto projektu se

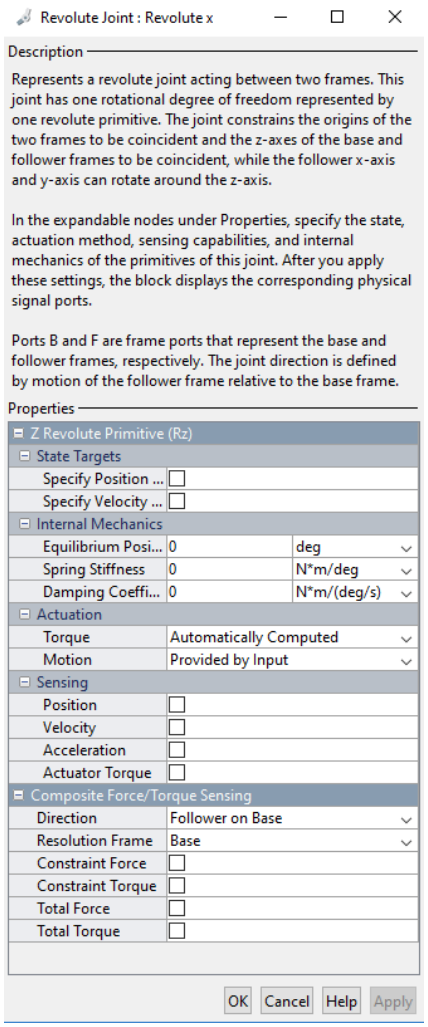
zabýváme hlavně bloky kloubů. Bloky objektů se základními vazbami jsou nahrány a propojeny už z přehrání modelu z Autodesk Inventoru. Bylo potřeba upravit klouby. Obzvláště kulové klouby. Tyto klouby nelze řídit jinak než pomocí momentů, avšak v této úloze neřešíme dynamiku, ale pouze dopřednou kinematiku, a tak byly tyto klouby nahrazeny třemi navzájem otočenými rotačními vazbami (viz obr. 22). Tyto klouby lze ovládat z hlediska pozice a hodí se pro naši úlohu.



Obrázek 22: Sférická vazba

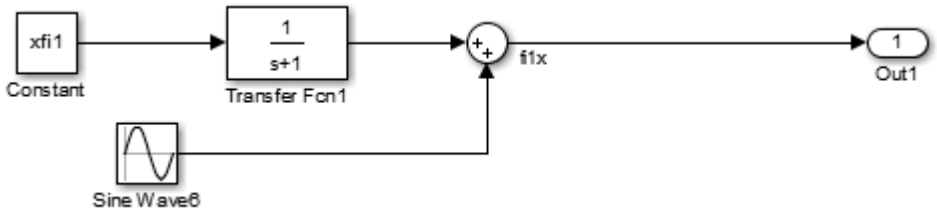
3.4.1.3 SOUŘADNICE POUŽITÉ K OVLÁDÁNÍ

K ovládání jednotlivých gest je zapotřebí dvou sad souřadnic, jedny pro zajištění polohy jednotlivých kloubů a druhé pro periodický pohyb při provádění gesta. K definici souřadnic byly použity soubory vytvořené v Matlabu přiřazující každému kloubu danou polohu a maximální výchylku při periodickém pohybu. Ovládání jednoho kloubu bylo zrealizováno následujícím způsobem.(viz obr. 23 a obr. 24)



Obrázek 23: Nastavení kloubu

Block Constant načítá souřadnici stacionární, tím se simulovaný kloub dostane do požadované polohy. Blok Sine Wave načítá souřadnici maximální výchylky a vytváří periodický pohyb kloubu. Blok Transfer Fcn vytváří přechodovou charakteristiku, která je potřeba pro plynulý pohyb, aby objekty neskákaly z jedné polohy do druhé, ale přesouvaly se plynule.



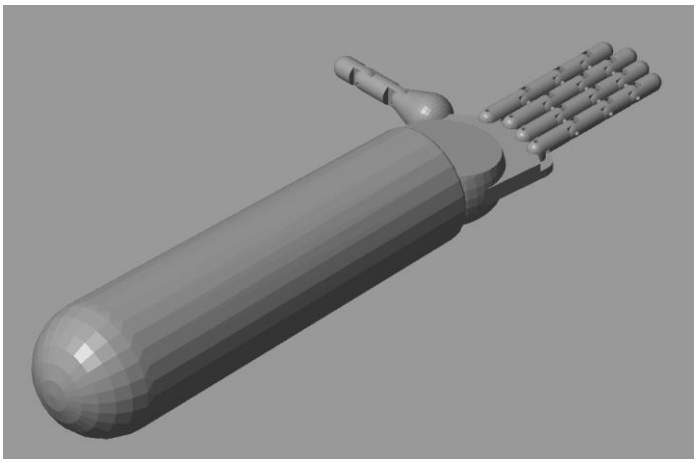
Obrázek 24:Ovládání kloubu

3.5 VÝSLEDNÁ GESTA

Gesta vytvořená pomocí simulace poměrně přesně aproximují gesta vytvářená pomocí lidské ruky.

3.5.1 INICIAČNÍ POLOHA

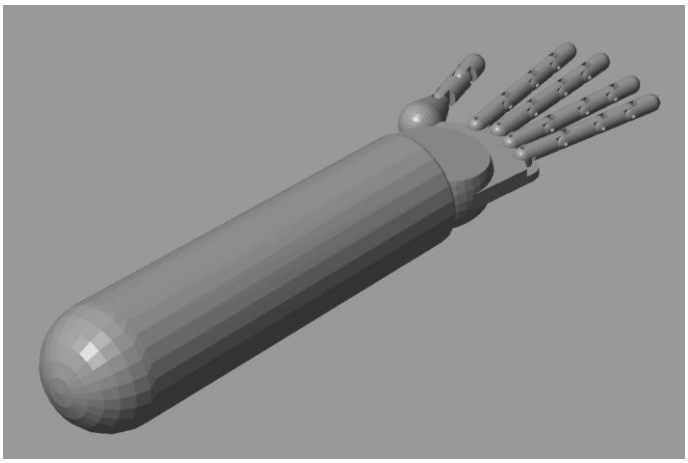
Iniciační poloha nastaví hodnotu ohybu všech kloubů na nulu (viz obr. 25).



Obrázek 25: Iniciační poloha

3.5.2 STARTOVACÍ GESTO

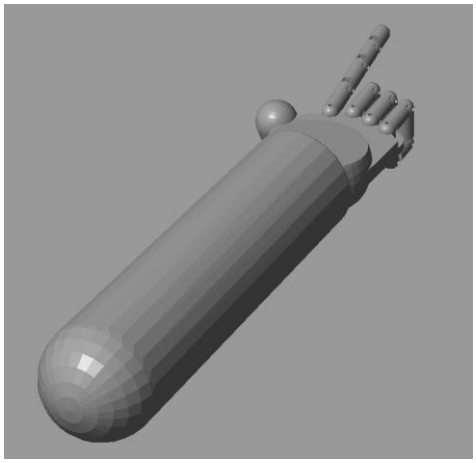
Startovací gesto se vytváří pomocí napřímené ruky s nataženými prsty, které jdou od sebe a k sobě. Tímto gestem systém infotainmentu rozpozná, že uživatel zahajuje ovládání gesty (viz obr. 26).



Obrázek 26: Startovací gesto

3.5.3 ONE FINGER GESTO

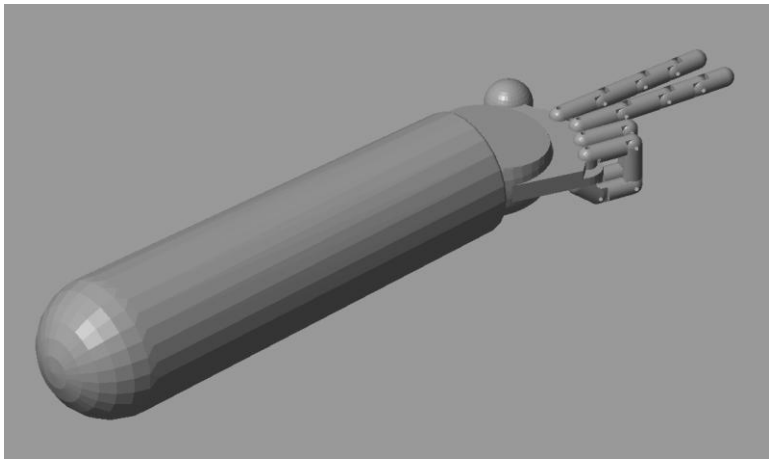
Toto gesto vytváří ruka natažená vpřed s prsty zaťatými v pěst s ukazováčkem nataženým ukazujícím na rádio v automobilu pohybujícím se zleva doprava a zpět. Tímto gestem se ovládá levé okénko nebo podbarvení palubní desky (viz obr. 27).



Obrázek 27: One finger gesto

3.5.4 TWO FINGER GESTO

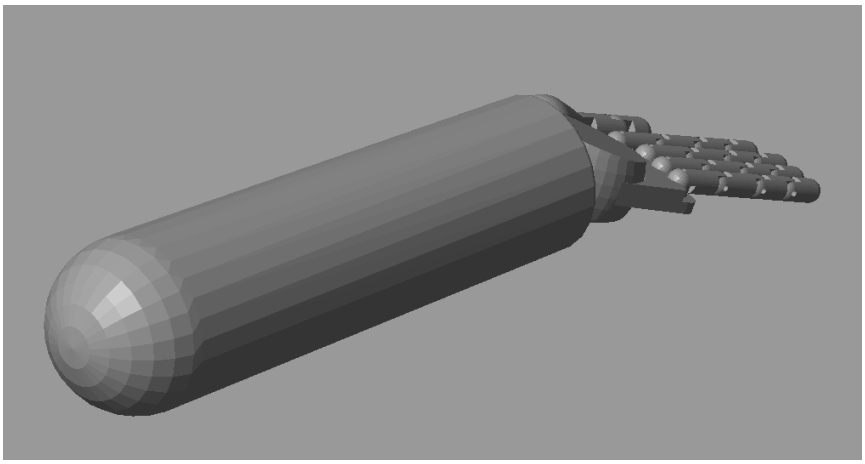
Tímto gestem se ovládá pravé okénko a druhá sada led diod v podbarvení palubní desky. Je podobné jako předchozí gesto One finger, ale oproti předchozímu je natažený i prostředníček (viz obr. 28).



Obrázek 28: Two finger gesto

3.5.5 WAVE GESTO

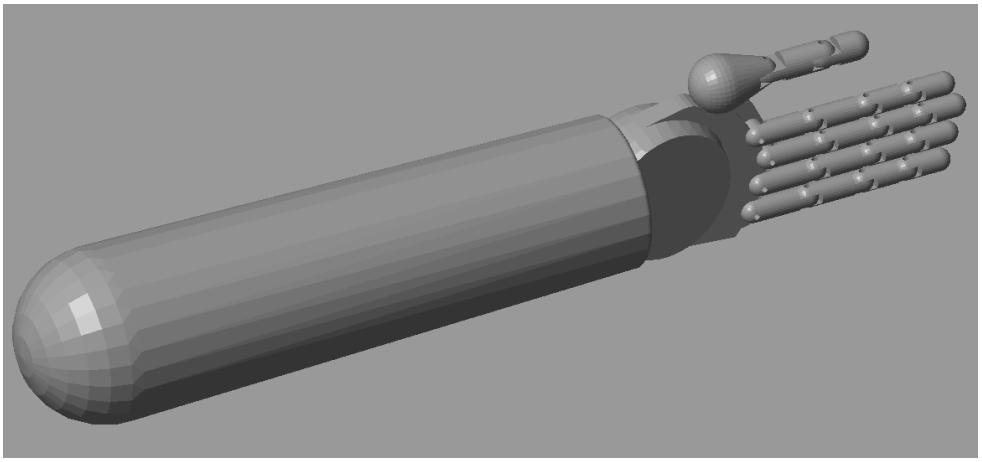
Toto gesto ovládá hlasitost rádia. Vytvoří se pomocí natažené ruky s nataženými prsty a pohybem nahoru a dolů (viz obr. 29).



Obrázek 29: Wave gesto

3.5.6 CHANGE SONG GESTO

Toto gesto ovládá přepínání skladeb. Pro provedení tohoto gesta se ruka přetočí palcem nahoru, natáhne prsty a pohybuje se z leva doprava a zpět (viz obr. 30).



Obrázek 30: Change song gesto

4 SOFTWAROVÉ INŽENÝRSTVÍ

Softwarové inženýrství je činnost, která zahrnuje informatiku, management a inženýrství. Cílem je tvorba, údržba softwarového díla. Jako i jiné inženýrské oblasti se i softwarové inženýrství zabývá hlavně spolehlivostí a cenou produktu.

V této práci bude uveden návrh softwarového díla..

4.1 DEKLARACE ZÁMĚRU PROJEKTU

Cílem projektu je vytvořit automatické testování senzoru v autě, který detekuje gesta lidské ruky. Pomocí gest se ovládají určité funkce auta (např. stahování okének, nastavování barvy interiéru, hlasitost radia atd.).

Z důvodu omezených zdrojů na tento projekt je výhodnější vytvořit simulaci pohybu modelu ruky prostřednictvím virtuální reality (simulinkového modelu), než vytvářet reálnou robotickou ruku mimikující pohyby ruky skutečné.

Jedná se o interakci mezi testerem, simulací a senzorem gest, kde tester zadá sled gest, která má simulace ruky udělat. Simulace vykoná sled gest, která budou snímána senzorem gest, pak se vyhodnotí a systém automobilu vykoná příslušné akce.

Jedná se o software pro testování používaný odborníkem. Není nutné vytvářet uživatelsky komfortní prostředí.

4.2 ODBORNÝ ČLÁNEK

Záměrem projektu je vytvoření simulace pro potřeby testování senzoru gest, který bude schopen zajistit věrohodnou simulaci pohybu lidské ruky. Dále je potřeba nechat kód otevřený pro další gesta, která se mohou přidat v dalších generacích senzoru.

1. Informace o natočení kloubů
   Tester si může nechat zobrazit informace o natočení kloubů a tím zjistit polohu všech částí simulované ruky. Stejně tak simulační model získává informace o natočení kloubů, které pak aplikuje do vizualizace.
2. Informace o délkách částí modelu
   Informace o geometrických parametrech modelu jsou k dispozici pouze programátorovi a vizualizaci.
3. Vizualizace
   Data z vizualizace budou k dispozici senzoru gest, testerovi i programátorovi.
4. Seznam gest
   Obsahuje seznam gest, ve kterém jsou zapsána natočení jednotlivých kloubů.

4.2.1 UŽIVATELSKÉ ROLE

Uživatelské role jsou znázorněny v kontextovém diagramu (viz obr. 31).

4.2.1.1 TESTER

- Dostává informace o natočení kloubů
- Má k dispozici seznam gest
- Dostává výstup ze simulace
- Dostává výstup povelů ze senzoru gest
- Zadává gesta ze seznamu gest

4.2.1.2 PROGRAMÁTOR

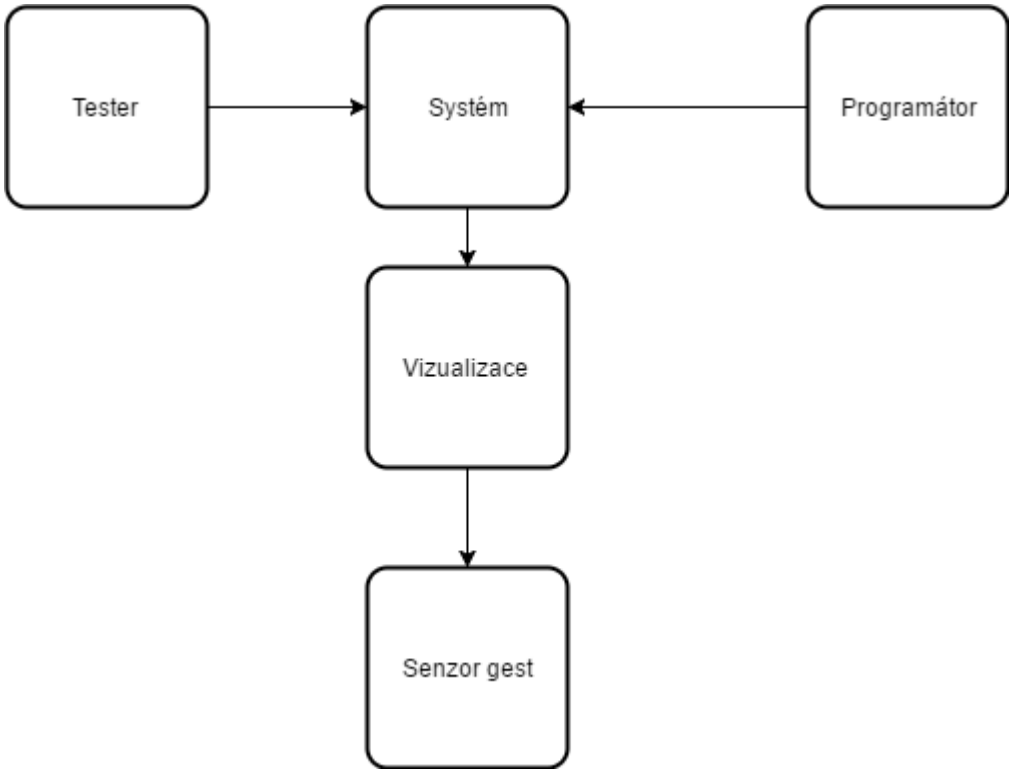
- Zadává informace o natočení do seznamu gest
- Má k dispozici délky částí modelu

4.2.1.3 VIZUALIZACE

- Dostává informace o natočení kloubu
- Vykresluje informace o natočení kloubu

4.2.1.4 SENZOR GEST

- Získává výstup ze simulace, který je převáděn do akcí prováděných ve vozidle.



Obrázek 31: Kontextový diagram

4.3 DATOVÝ SLOVNÍK

V datovém slovníku je popsána struktura databáze, jednotlivé tabulky a jejich položky. Následně je popsána struktura na schématu toku dat (viz obr. 32).

4.3.1 GESTURE

Tabulka obsahuje informace o aktivovaných gestech (IDGesture), jednotlivých úhlech (IDangle) a hodnotách jakých mají úhly nabývat.

Jméno	Typ
IDgesture	String
IDangle	String
SteadyAngle	Integer
Amplitude	Integer

4.3.2 MODEL

Tabulka modelu obsahuje informace o jednotlivých tělesech, jejich rozměrech, poloze a o kloubech a jejich pozicích.

Jméno	Typ
IDbody	String
BotyParameters	Integer
BodyPosition	Array
IDJoint	String
JointPosition	Integer

4.3.3 SIMULATION

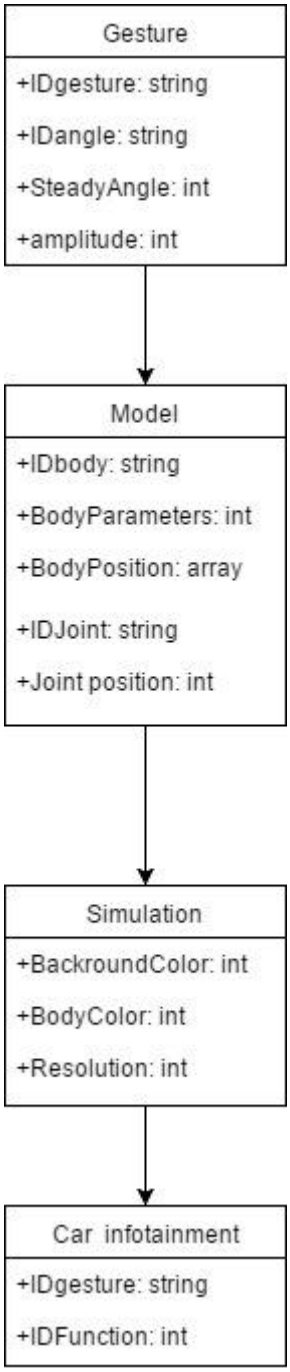
Tabulka udává informace o simulaci. Jaká bude barva pozadí, barva jednotlivých těles a jaké bude mít simulace rozlišení.

Jméno	Typ
BackroundColor	Intenger
BodyColor	Integer
Resolution	Integer

4.3.4 CAR INFOTAINMENT

Tabulka udává informace o zjištěných gestech a funkcích, které mají spustit.

Jméno	Typ
IDgesture	String
IDFunction	Integer



Obrázek 32: Data flow

4.4 ŘEŠITELSKÝ TÝM

- Software architekt
- Databázový specialista
- Matlab specialista
- Mechatronik
- CAN specialista
- 3 programátoři C

4.5 NÁVRH ŘEŠENÍ A ROZPOČET

Pro úspěšnou implementaci projektu je zapotřebí minimálně jednoho počítače, přibližná cena 30 000 Kč, s licencí Matlabu se Simulinkem s toolboxem Simechanics, přibližná cena 500 000 Kč, větší monitor, přibližná cena 5 000 Kč, a infotaiment vozu Volkswagen Golf R touch ten bude zapůjčen zákazníkem.

Cena HW (30 000 + 5 000 = 35 000Kč)

Cena SW (500 000Kč)

4.5.1 COCOMO

Doba vývoje v člověkoměsících podle COCOMO, vázaný mód – pracuje se na SW i HW současně, ladění čteček karet, odbavovacích terminálů, apod. V závislosti na komplexnosti problému, který je v této práci řešen je počet řádek kódu odhadnut na 5000 řádků kódu. Byla navrhnuta cena práce programátora na 5000 $ na měsíc práce. Do této ceny se nezapočítává pouze plat programátora, ale i náklady na provoz programátora. Výpočet byl proveden pomocí online aplikace na stránce, zde http://csse.usc.edu/tools/cocomoii.php (viz obr. 33)

Výsledná cena je uvedena v dolarech, tedy 93030 $, v české měně se pohybujeme okolo 2 267 000 Kč.

4.5.2 VÝSLEDNÁ CENA

Když sečteme všechny náklady, tak se dostaneme přibližně na částku 2 800 000. Za firemní náklady. Zákazníkovi bude nabídnuta cena podle cenové politiky firmy.

Software Size Sizing Method: Source Lines of Code

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	5000						
Reused		0	0				
Modified							

Software Scale Drivers

Precedentedness	Very Low	Architecture / Risk Resolution	Very Low	Process Maturity	Very Low
Development Flexibility	Very Low	Team Cohesion	Very High		

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	Nominal	Analyst Capability	Nominal	Time Constraint	High
Data Base Size	Nominal	Programmer Capability	Nominal	Storage Constraint	Nominal
Product Complexity	Nominal	Personnel Continuity	Nominal	Platform Volatility	Nominal
Developed for Reusability	Nominal	Application Experience	Nominal	**Project**	
Documentation Match to Lifecycle Needs	Nominal	Platform Experience	Nominal	Use of Software Tools	Very High
		Language and Toolset Experience	Low	Multisite Development	Nominal
				Required Development Schedule	Nominal

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars) 5000

Calculate

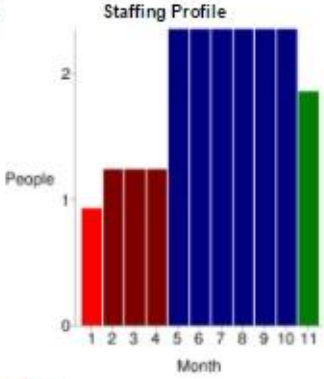
Results

Software Development (Elaboration and Construction)

Effort = 18.6 Person-months
Schedule = 9.6 Months
Cost = $93030

Total Equivalent Size = 5000 SLOC

Staffing Profile



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	1.1	1.2	0.9	$5582
Elaboration	4.5	3.6	1.2	$22327
Construction	14.1	6.0	2.3	$70703
Transition	2.2	1.2	1.9	$11164

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.2	0.5	1.4	0.3
Environment/CM	0.1	0.4	0.7	0.1
Requirements	0.4	0.8	1.1	0.1
Design	0.2	1.6	2.3	0.1
Implementation	0.1	0.6	4.8	0.4
Assessment	0.1	0.4	3.4	0.5
Deployment	0.0	0.1	0.4	0.7

Your output file is http://csse.usc.edu/tools/data/COCOMO_May_14_2017_10_42_48_454460.txt

Created by Ray Madachy at the Naval Postgraduate School. For more information contact him at rjmadach@nps.edu

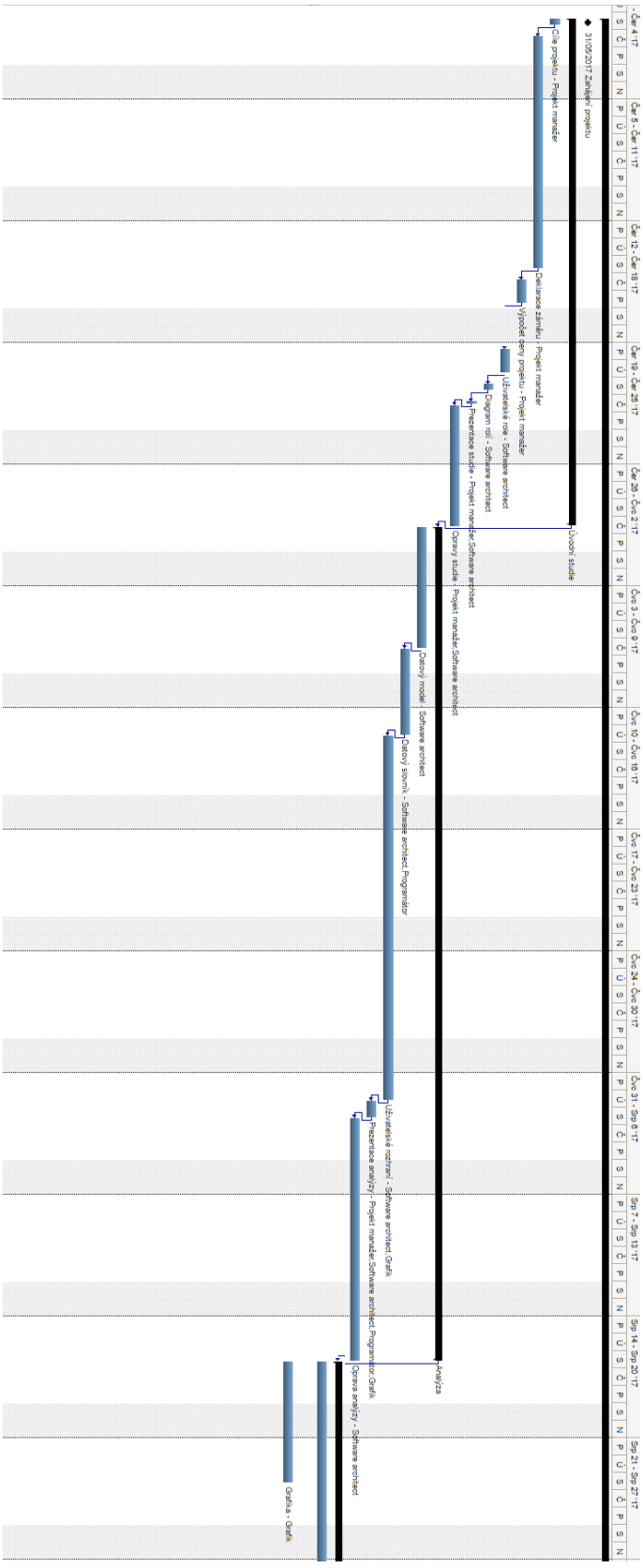
Obrázek 33: Výsledky výpočtu COCOMO

4.6 ČASOVÝ HARMONOGRAM

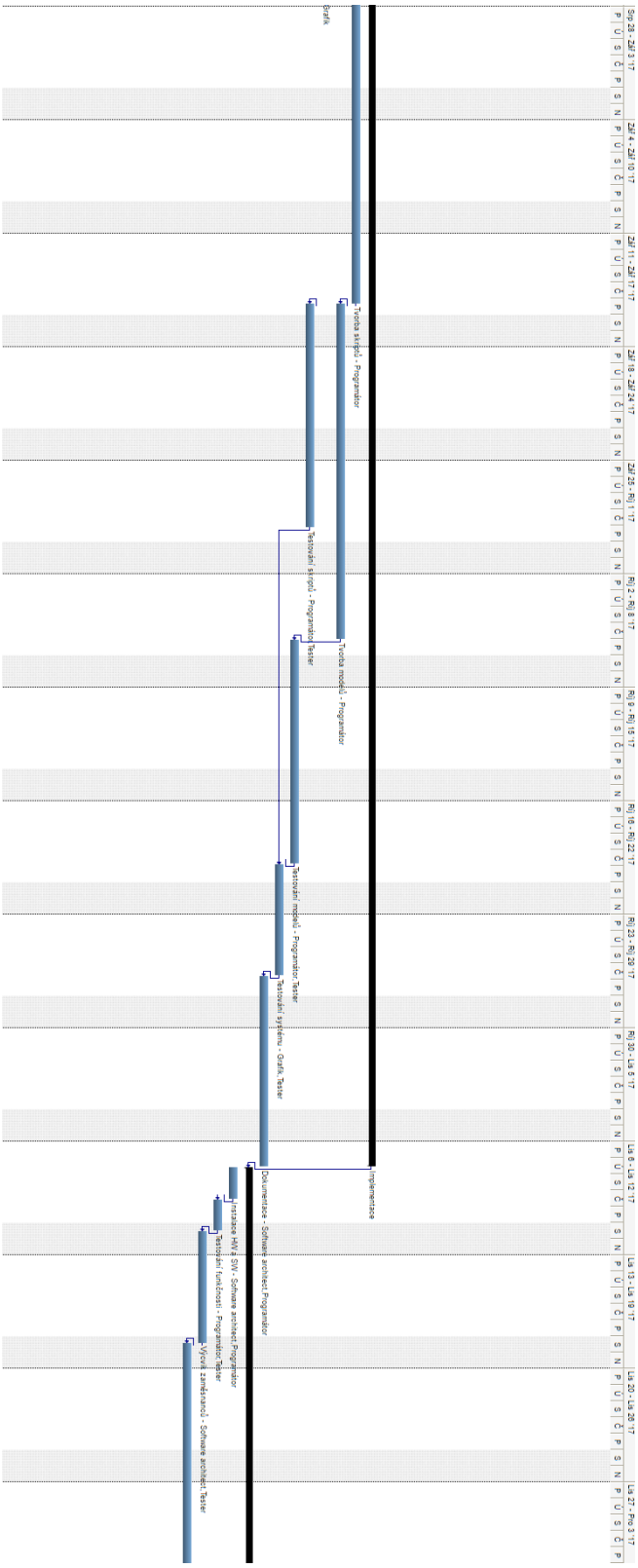
Časový harmonogram s rozdělením zdrojů je vytvořen pomocí ganttova diagramu (obr. 34-37).

Název	Trvání
Zahájení projektu	**0d**
⊟Úvodní studie	**21.5d**
Cíle projektu	1d
Deklarace záměru	10d
Výpočet ceny projektu	2d
Uživatelské role	2d
Diagram rolí	1d
Prezentace studie	0.5d
Opravy studie	5d
⊟Analýza	**34d**
Datový model	5d
Datový slovník	3d
Uživatelské rozhraní	15d
Prezentace analýzy	1d
Oprava analýzy	10d
⊟Implementace	**58d**
Tvorba skriptů	20d
Tvorba modelů	15d
Grafika	5d
Testování skriptů	10d
Testování modelů	10d
Testování systému	5d
Dokumentace	8d
⊟Zavedení systému	**29d**
Instalace HW a SW	2d
Testování funkčnosti	2d
Výcvik zaměsnanců	5d
Testování a opravy	20d
Ukončení	0d

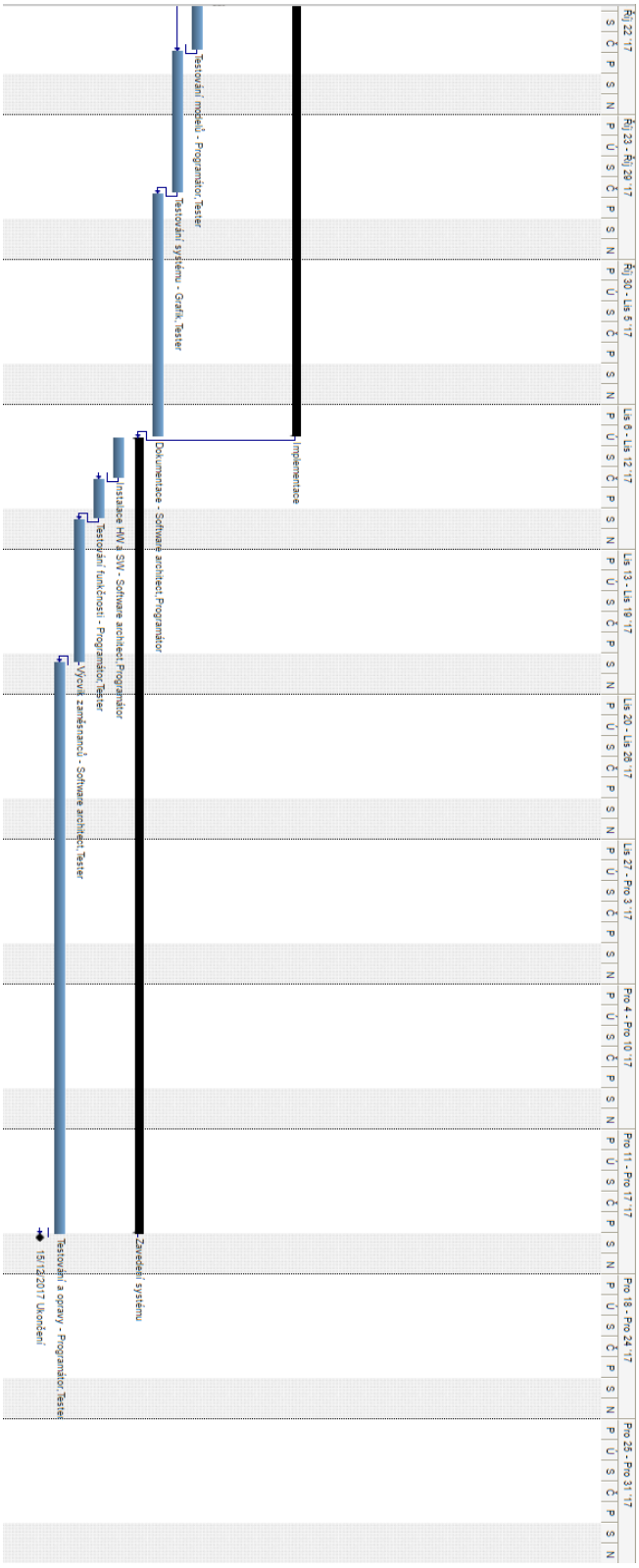
Obrázek 34: Tabulka časového harmonogramu



Obrázek 35:Ganttův diagram část 1



Obrázek 36: Ganttův diagram část 2



Obrázek 37: Ganttův diagram část 3

5 VÝPOČET INVERZNÍ KINEMATIKY

Výpočet inverzní kinematiky bude v této kapitole řešen pomocí optimalizačních metod. Výpočetní model bude sestaven v kapitole 5.1.2. Tento model byl sestaven maticovou metodou.

5.1 MATICOVÁ METODA

Maticová metoda slouží k řešení problematiky syntézy kinematických a dynamických úloh složitějších prostorových kinematických řetězců. [4]

Tato metoda slouží k popisu kinematiky mechanického řetězce. Jelikož máme otevřený dopředný kinematický řetězec, vede tato metoda k popisu vztahu mezi polohou endefektoru (ruky) a motorovými souřadnicemi jednotlivých aktuovaných vazeb mechanismu. Pomocí řetězení matic popíšeme transformace z jednoho tělesa do druhého a tím popisujeme významné body v mechanismu. Matice ve 2D mají tvar popsaný v kapitole 5.1.1.

5.1.1 MATICOVÁ METODA VE 2D

$$\begin{bmatrix} 1 & 0 & X \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Tato matice popisuje posuv ve směru x o X.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & Y \\ 0 & 0 & 1 \end{bmatrix}$$

Tato matice popisuje posuv ve směru y o Y.

$$\begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Tato matice popisuje rotaci okolo osy z o úhel φ.

Podobně můžeme použít matice i pro náš případ. Ale na rozdíl od matic výše uvedených je naše úloha ve 3D, tím pádem je nutné matice rozšířit o jednu dimenzi. Počet matic a základních pohybů se tu zvýší na 6 (viz kapitola 5.1.2).

5.1.2 MATICOVÁ METODA VE 3D

Využití matic této metody v Matlabu byly matice přepsány do funkcí.

5.1.2.1 TRANSLACE VE SMĚRU X

```
function matice_Tx=Tx(x)

matice_Tx = [1 0 0 x
0 1 0 0
0 0 1 0
0 0 0 1];

end
```

5.1.2.2 TRANSLACE VE SMĚRU Y

```
function matice_Ty=Ty(y)

matice_Ty = [1 0 0 0
      0 1 0 y
      0 0 1 0
      0 0 0 1];

end
```

5.1.2.3 TRANSLACE VE SMĚRU Z

```
function matice_Tz=Tz(z)

matice_Tz = [1 0 0 0
             0 1 0 0
             0 0 1 z
             0 0 0 1];

end
```

Jako další transformační matice jsou nutné rotace.

5.1.2.4 ROTACE OKOLO OSY X

```
function matice_Txfi=Txfi(xfi)

matice_Txfi = [1     0       0      0
               0 cos(xfi) -sin(xfi) 0
               0 sin(xfi)  cos(xfi) 0
               0     0       0      1];

end
```

5.1.2.5 ROTACE OKOLO OSY Y

```
function matice_Tyfi=Tyfi(yfi)
```

```
matice_Tyfi = [cos(yfi)      0       sin(yfi)      0
                   0         1         0           0
               -sin(yfi)     0       cos(yfi)      0
                   0         0         0           1];

end
```

5.1.2.6 ROTACE OKOLO OSY Z

```
function matice_Tzfi=Tzfi(zfi)

matice_Tzfi = [cos(zfi) -sin(zfi) 0 0
        sin(zfi)  cos(zfi) 0 0
        0          0       1 0
        0          0       0 1];

end
```

5.1.3 POUŽITÍ VÍCE TRANFORMACÍ

Pro použití více transformací se transformační matice mezi sebou násobí. Proto byl napsán skript, který transformuje souřadnice z počátku souřadnic do konečků prstů.

```
%% zapesti
T12=Txfi(xfi1)*Tyfi(yfi1)*Tzfi(zfi1)*Tx(a);

%% palec
T23=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x23)*Ty(y23);
T34=Tyfi(yfi3)*Tzfi(zfi3)*Txfi(xfi3)*Tx(b);
T45=Tzfi(zfi4)*Tx(c);
T56=Tzfi(zfi5)*Tx(d);

%% ukazovacek
T27=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x27)*Ty(y27);
T78=Tzfi(zfi7)*Tx(e);
T89=Tyfi(yfi8)*Tx(f);
T910=Tyfi(yfi9)*Tx(g);
T1011=Tyfi(yfi10)*Tx(h);

%% prostrednicek
T212=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x212)*Ty(y212);
T1213=Tzfi(zfi12)*Tx(i);
T1314=Tyfi(yfi13)*Tx(j);
T1415=Tyfi(yfi14)*Tx(k);
T1516=Tyfi(yfi15)*Tx(l);

%% prstenicek
T217=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x217)*Ty(y217);
T1718=Tzfi(zfi17)*Tx(m);
T1819=Tyfi(yfi18)*Tx(n);
T1920=Tyfi(yfi19)*Tx(o);
T2021=Tyfi(yfi20)*Tx(p);
```

```
%% malicek
T222=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x222)*Ty(y222);
T2223=Tzfi(zfi22)*Tx(q);
T2324=Tyfi(yfi23)*Tx(r);
T2425=Tyfi(yfi24)*Tx(s);
T2526=Tyfi(yfi25)*Tx(t);

%% transformace

Tpalec=T12*T23*T34*T45*T56;
Tukazovacek=T12*T27*T78*T89*T910*T1011;
Tprostrednicek=T12*T212*T1213*T1314*T1415*T1516;
Tprstenicek=T12*T217*T1718*T1819*T1920*T2021;
Tmalicek=T12*T222*T2223*T2324*T2425*T2526;
```

5.1.4 SOUŘADNICE

Souřadnice xpalec, xukazovacek, xprostrednicek, xprstenicek a xmalicek jsou souřadnice bodu v konečcích prstů.

```
x0=[0
    0
    0
    1];

xpalec=Tpalec*x0;
xukazovacek=Tukazovacek*x0;
xprostrednicek=Tprostrednicek*x0;
xprstenicek=Tprstenicek*x0;
xmalicek=Tmalicek*x0;
```

Pro další postup je nutné definovat i následující souřadnice. Tyto souřadnice použijeme při optimalizaci, kdy je nutné mít přeurčenou úlohu, aby optimalizační algoritmus nalezl nejvíce vyhovující polohu mechanismu.

```
xtop=T12*x0;
```

Dále je nutné definovat souřadnice, na které chci konečky prstů dostat. Jedná se o polohu pro úchop válcovitého předmětu. Prsty se musí dostat do definovaných bodů na kružnici. Přesný popis souřadnic je popsán v příloze 8.2.2. maticovy_zapis. Tuto polohu se algoritmus bude snažit dosáhnout při optimalizaci dalších podmínek definovaných v kapitole 5.2.1.

```
%% kruh

stred=[625, 0, 0];

polomer=50;
```

```
palecSouradnice=[stred(1)+polomer*cos(pi)
                 stred(2)+polomer*sin(pi)
                 0];
ukazovacekSouradnice=[stred(1)+polomer*cos(-90*pi/180)
                      stred(2)+polomer*sin(-90*pi/180)
                      0];
prostrednicekSouradnice=[stred(1)+polomer*cos(0)
                       stred(2)+polomer*sin(0)
                       0];
prstenicekSouradnice=[stred(1)+polomer*cos(60*pi/180)
                     stred(2)+polomer*sin(60*pi/180)
                     0]';
malicekSouradnice=[stred(1)+polomer*cos(120*pi/180)
                   stred(2)+polomer*sin(120*pi/180)
                   0]';
```

5.2 OPTIMALIZACE

Chceme-li nalézt optimalizační parametry v našem případě $fi = [\varphi_1 \dots \varphi_{27}]$, to jsou souřadnice natočení jednotlivých kloubů v daných směrech, přesně popsáno v příloze 8.2.2 maticovy_zapis. Hledáme vektor fi nezávislých parametrů, pro který nabývá cílová funkce extrému. Podmínky ve funkci musí být dány tak, aby vznikl přeurčený systém. Pouze přeurčený systém lze optimalizovat. Pro tuto práci je přeurčení systému zajištěno minimalizací fi a připojením dalších podmínek popsaných v kapitole 5.2.1.

5.2.1 CÍLOVÁ FUNKCE

Cílová funkce popisuje kvantitativně míru stavu, kterého chceme dosáhnout. Tato funkce je důležitá s ohledem na výběr vhodné metody a musí být analytickou funkcí parametrů.

V našem případě cílová funkce popisuje vzdálenost mezi konečky prstů a body na kružnici. Avšak pouze tato podmínka nedefinuje jen jednu optimální polohu, je nutné připojit ještě další podmínky, tak aby vznikl přeurčený systém. Jednou z dalších podmínek je, že změny úhlů by měli být minimální a zápěstí by se mělo dostat do dané výšky.

Tato funkce je zapsaná ve skriptu.

```
%fitness fce

function y = myFitness(fi)
maticovy_zapis;



rlx=abs(fi(1)-0);
```

```
rl=rlx;

firovnom=(abs(fi(1))+abs(fi(2))+abs(fi(3))+abs(fi(4))+abs(fi(5))+abs(fi(6))
+abs(fi(7))+abs(fi(8))+abs(fi(9))+abs(fi(10))+abs(fi(11))+abs(fi(12))+abs(f
i(13))+abs(fi(14))+abs(fi(15))+abs(fi(16))+abs(fi(17))+abs(fi(18))+abs(fi(1
9))+abs(fi(20))+abs(fi(21))+abs(fi(22))+abs(fi(23))+abs(fi(24))+abs(fi(25))
+abs(fi(26))+abs(fi(27)));

dlantopz=abs(xtop(3)-100);
dlantop=dlantopz;


vzdalenostPalec=(((xpalec(1)-palecSouradnice(1))^2)+((xpalec(2)-
palecSouradnice(2))^2)+(xpalec(3)-palecSouradnice(3))^2)^1/2;

vzdalenostUkazovacek=(((xukazovacek(1)-
ukazovacekSouradnice(1))^2)+((xukazovacek(2)-
ukazovacekSouradnice(2))^2)+((xukazovacek(3)-
ukazovacekSouradnice(3))^2))^1/2;

vzdalenostProstrednicek=((xprostrednicek(1)-
prostrednicekSouradnice(1))^2+(xprostrednicek(2)-
prostrednicekSouradnice(2))^2+(xprostrednicek(3)-
prostrednicekSouradnice(3))^2)^1/2;

vzdalenostPrstenicek=((xprstenicek(1)-
prstenicekSouradnice(1))^2+(xprstenicek(2)-
prstenicekSouradnice(2))^2+(xprstenicek(3)-prstenicekSouradnice(3))^2)^1/2;

vzdalenostMalicek=((xmalicek(1)-malicekSouradnice(1))^2+(xmalicek(2)-
malicekSouradnice(2))^2+(xmalicek(3)-malicekSouradnice(3))^2)^1/2;

y = rl + 1*firovnom + 100*dlantop +
10000*(vzdalenostPalec+vzdalenostUkazovacek+vzdalenostProstrednicek+vzdalen
ostPrstenicek+vzdalenostMalicek);



end
```

5.3 METODY LOKÁLNÍ A GLOBÁLNÍ OPTIMALIZACE

5.3.1.1 LOKÁLNÍ OPTIMALIZACE

Lokální optimalizace provedeme analyticky. Je potřeba nalézt extrém funkce F. Extrém lze nalézt pomocí rovnice (1).

$$\left(\frac{\partial F}{\partial x}\right)_{x^*} = F'(x^*) = 0 \qquad (2)$$

Tento bod nazýváme bodem podezřelým z extrému. Pro ověření se počítá druhá derivace a zjišťuje se její znaménko. Podle ní se rozhoduje, zda jde o extrém či nikoliv.

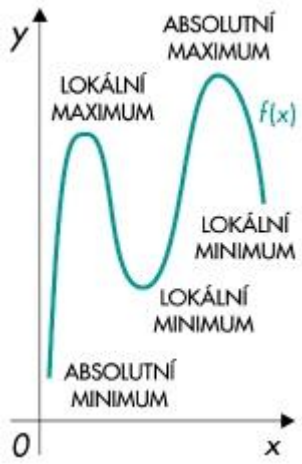
$$F''(x^*) > 0 \quad \rightarrow \text{je extrém (minimum)} \tag{3}$$

$$F''(x^*) < 0 \rightarrow \text{je extrém (maximum)} \tag{4}$$

$$F''(x^*) = 0 \rightarrow \text{není extrém} \tag{5}$$

Pomocí těchto výpočtů získáme množinu lokálních extrémů. Tyto výpočty můžeme provést exaktně.

Jelikož tato metoda nenajde pouze globální extrémy (obr. 38), je zapotřebí použít metody globální optimalizace.



Obrázek 38: Lokální a globální extrém [5]

5.3.1.2 GLOBÁLNÍ OPTIMALIZACE

Analytický postup pro globální optimalizaci je prováděn tak, že z množiny lokálních extrémů vyberu ty globální tím, že porovnám množinu lokálních extrémů a vyberu z ní bod, který má nejvyšší hodnotu pro globální maximum a nejnižší hodnotu pro globální minimum. Pokud se jedná o jediný bod, mohu bod nazvat ostrým globálním maximem (minimem). Tento algoritmus je neefektivní, hlavně pokud se jedná o funkci více proměnných. Proto se pro optimalizaci používají další metody jako jsou genetické algoritmy nebo metoda simulovaného žíhání.

5.3.2 GENETICKÝ ALGORITMUS

Jednou z často používaných metod globální optimalizace jsou genetické algoritmy. Jejich cílem je nalézt globální extrém. Volba prohledávané oblasti je pro tuto práci výhodná kvůli možnosti nastavení maximálních poloh kloubů. Z hlediska sestavení optimalizovaného algoritmu sledujeme takzvanou populaci, což je sada bodů v prostoru parametrů. Vývoj populace je zajištěn právě genetickým algoritmem. Výchozí populací nazveme P_0, tato populace má N bodů „jedinců“.

5.3.2.1 ALGORTMUS

0. Výchozí populace P_0
1. i+1=i
2. Výběr funkce $P_{i-1} \rightarrow P_i' \Rightarrow$ To je rodičovská populace což jsou jedinci, kteří se otisknou do další generace.
3. Reprodukční funkce $(kP_i') \rightarrow \overline{P_i}$ (k je funkce křížení)
4. Náhodné mutace $\overline{P_i} \rightarrow P_i$
5. Kritérium ukončení

Pokud je splněno kritérium ukončení, pak metoda ukončí výpočet, pokud není, tak se vracíme na bod 1.

Genetické algoritmy obvykle maximalizují cílovou funkci.

O míře vlivu rodičů rozhoduje cílová funkce, ale ne absolutně. Svou roli tu mají i pravděpodobnostní hlediska. Všechna pravidla výběru popisují výběrové funkce.

5.3.2.1.1 Křížení

Reprodukční funkce popisují křížení rodičů $\overrightarrow{x'}$ a $\overrightarrow{y'}$ a vznik následné generace $\vec{x}$ a $\vec{y}$.

Aritmetické křížení probíhá za pomoci náhodného váhového koeficientu $r \in \varphi(0;1)$ tak, že platí (6) a (7).

$$\vec{x} = \mathrm{r}\overrightarrow{x'} + (1-\mathrm{r})\overrightarrow{y'} \tag{6}$$

$$\vec{y} = \mathrm{r}\overrightarrow{y'} + (1-\mathrm{r})\overrightarrow{x'} \tag{7}$$

Dalším možným křížením je jednoduché křížení, pro něž platí (8).

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_M \end{pmatrix} \quad \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_M \end{pmatrix} \quad \vec{x}^E = \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ y_{k+1} \\ \vdots \\ y_M \end{pmatrix} \quad \vec{y}^E = \begin{pmatrix} y_1 \\ \vdots \\ y_k \\ x_{k+1} \\ \vdots \\ x_M \end{pmatrix} \tag{8}$$

M je zde dimenze prostoru parametrů a $k \leq r \leq k+1$, kde $r \in \varphi\langle 1; M\rangle$.

5.3.2.1.2 Mutace

Mutace mění jeden nebo více parametrů jedince. Výsledkem může být zanesení nové genetické informace do jedince. S použitím mutací je možné u genetického algoritmu zabránit naplnění kritéria ukončení v lokálním optimu. Využití je dáno na základě pravděpodobnostní hodnoty výskytu mutace, kterou si uživatel může zvolit.

5.3.2.1.3 Kritéria ukončení

Jedním z kritérií může být maximální počet generací, další je hodnota cílové funkce, případně změna její hodnoty. Pokud změna cílové funkce klesne pod danou hodnotu je splněno kritérium ukončení. Toto kritérium bylo využito i v našem případě.

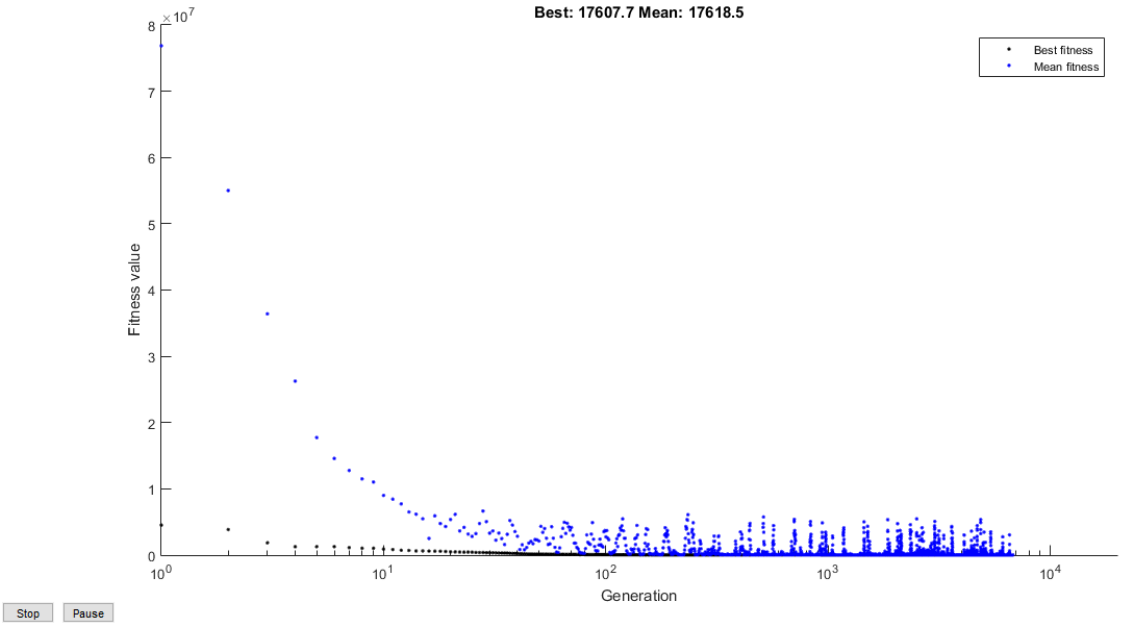
5.3.2.2 VÝPOČET

Výpočet v našem případě byl proveden pomocí funkce ga, která je předprogramována v Matlabu. Meze byly stanoveny skripty LB a UB, přiloženy v přílohách. Počet parametrů je 27. Cílovou funkci jsme si definovali v kapitole 5.2.1. Funkce ga hledá minimum cílové funkce a není třeba ji modifikovat.

```
% main function
clc
close all,
init;
LB;
UB;
fitfce=@myFitness;
nvars=27;
options = gaoptimset('Display','iter',... %display every iteration
'Generations',20000,... %maximum number of generations is 70
'TolFun',1e-7,... %tolerance for optimisation is 0.01
'TolCon',1e-7,...
'PlotFcns',@gaplotbestf);

[fi,fval,exitflag,output,population]=ga(fitfce, nvars, [],[],[],[],lb,
ub,[], options);
fi
```

5.3.2.3 VÝSEDEK



Obrázek 39: graf genetického algoritmu

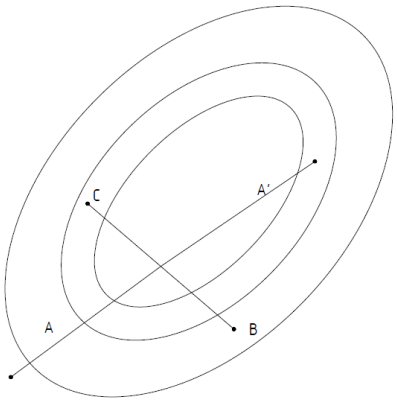
Pro hledání globálního optima je vhodné použít metodu hledání lokálního optima. Výpočet je proto následně zpřesněn (kapitola 5.3.3.2.2) funkcí fminsearch, která začíná na výsledcích výpočtu funkce ga. Výsledná hodnota cílové funkce po 14141 generacích je 17607,7 (viz obr. 39).

5.3.3 SIMPLEXOVÁ METODA

Simplexová metoda je metoda k nalezení lokálního optima. Simplex je v podstatě objekt, který má n+1 vrcholů. Dimenzi problému značíme n.

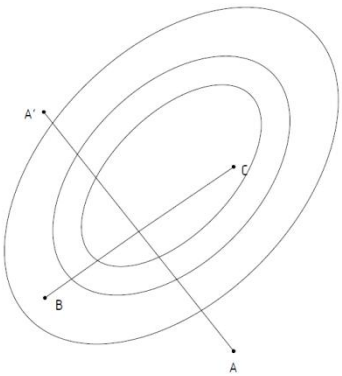
V simplexové metodě se řídíme několika pravidly:

1. pravidlo – vypustíme vrchol s nejhorší hodnotou cílové funkce F a nahradíme novým nezrcadleným vrcholem na spojnici vypuštěným a střediska vrcholů zbývajících (viz obr. 40).



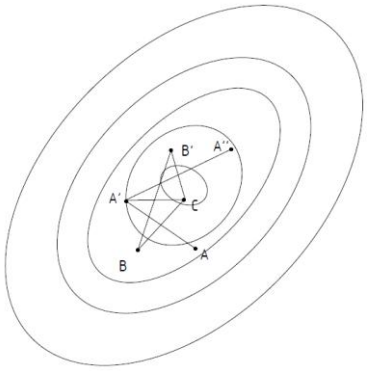
Obrázek 40:1. pravidlo simplexové metody

2. pravidlo – nelze se vracet do právě vypuštěného vrcholu. Pokud má nový vrchol nejhorší hodnotu F, pak se vypouští druhý nejhorší (viz obr. 41).



Obrázek 41: 2. pravidlo simplexové metody

3. pravidlo – jedině vrchol s nejlepší hodnotou F zůstává na místě více než n výměn, redukuje se hrana simplexu (viz obr. 42)



Obrázek 42: 3. pravidlo simplexové metody

5.3.3.1 VÝPOČET

Výpočet byl proveden pomocí funkce fmincon, která je předprogramována v Matlabu. Meze byly stanoveny skripty LB a UB a jsou přiloženy v přílohách. Počet optimalizovaných parametrů je 27. Cílovou funkci jsme definovali v kapitole 5.2.1. Počáteční hodnoty byly definovány jako nulové.

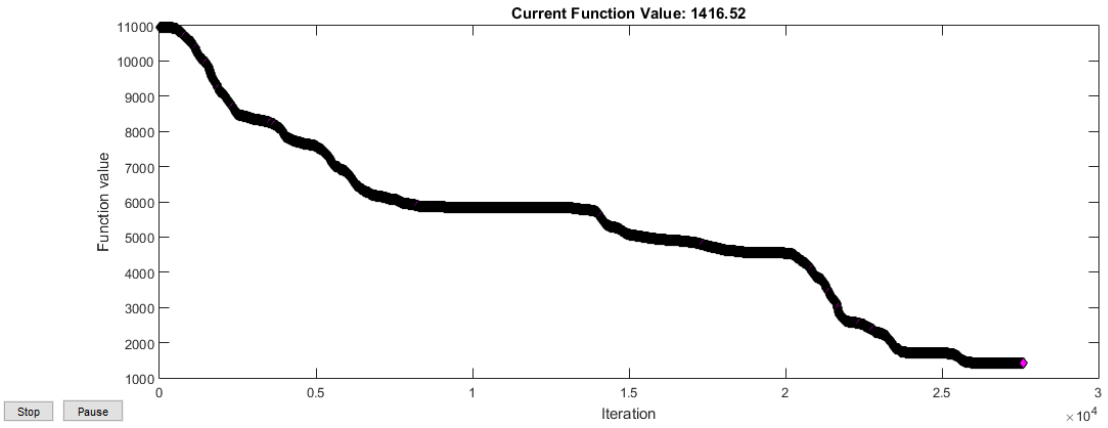
```
% main function


close all,

fi=zeros(27,1);
LB;
UB;
fitfce=@myFitness;
options
=optimset('display','iter','MaxIter',100000,'MaxFunEval',5000000,'TolX',1e-
8,'TolFun',1e-8,'PlotFcns',@optimplotfval);
[fi,fval,exitflag,output,lambda,grad,hessian] =
fmincon(fitfce,fi,[],[],[],[],lb,ub,[],options);
```

5.3.3.2 VÝSLEDKY

5.3.3.2.1 S využitím funkce fminsearch



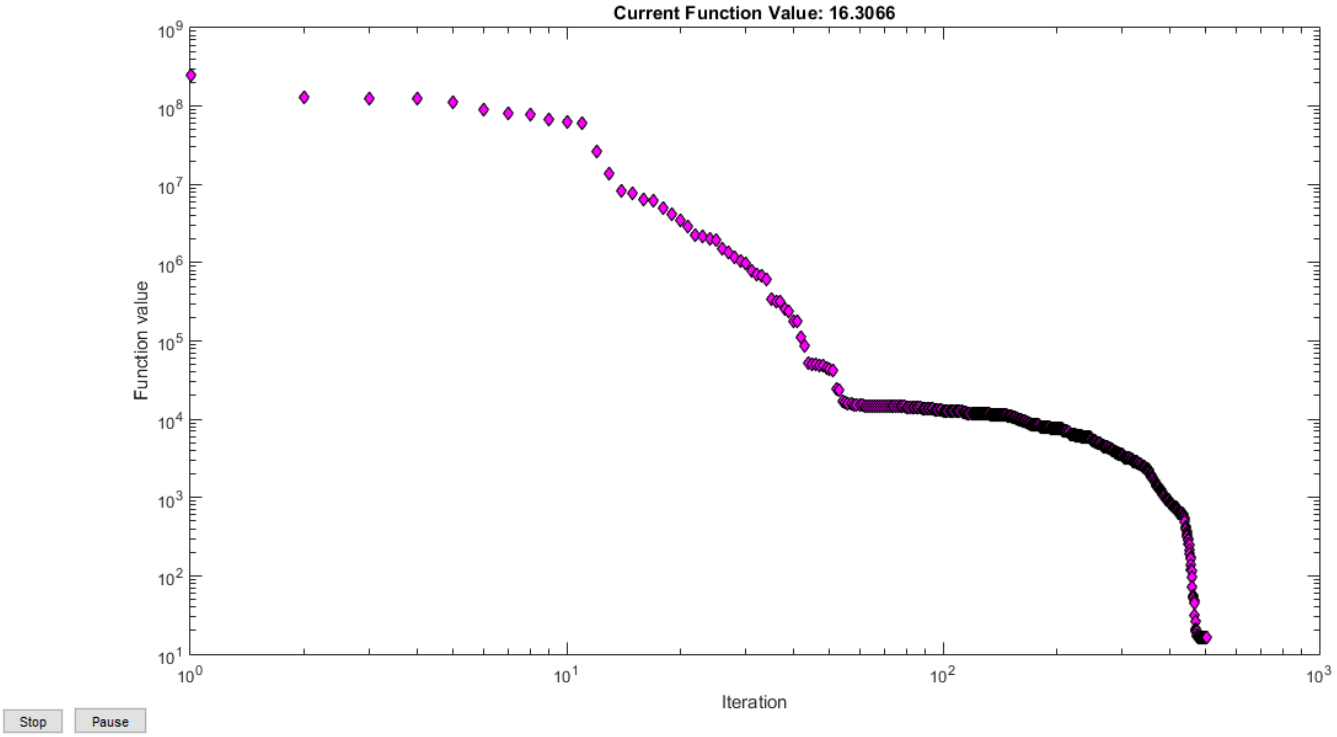
Obrázek 43: graf výpočtu fminsearch

Výsledkem výpočtu inverzní kinematiky pomocí funkce fminsearch, která navazuje na výpočet globální optimalizace funkcí ga, je vektor natočení fi, ze kterého dopočítáme polohu a rozdíl polohy od žádané. Výsledná hodnota cílové funkce 1416,52. Výpočet konvergoval po 27612 iteracích (viz obr. 43). Výpočet trval i s výpočtem ga 12,577 hodin.

```
xpalec = [5.750e+02;0.0011;0.0014;1]
xukazovacek = [6.603e+02;-35.351;-8.17e-04;1]
xprostrednicek [6.749e+02;2.81e-04;7.94e-04;1]
xprstenicek = [6.5001e+02;43.27;0.0131;1]
xmalicek = [5.9999e+02;43.343;-0.022;1]
```

```
vzdalenostPalec =  1.7937e-06 mm
vzdalenostUkazovacek =  1.0234e-05 mm
vzdalenostProstrednicek =  9.0302e-07 mm
vzdalenostPrstenicek =  5.9126e-04 mm
vzdalenostMalicek =  0.0012 mm
```

5.3.3.2.2 S použitím fmincon

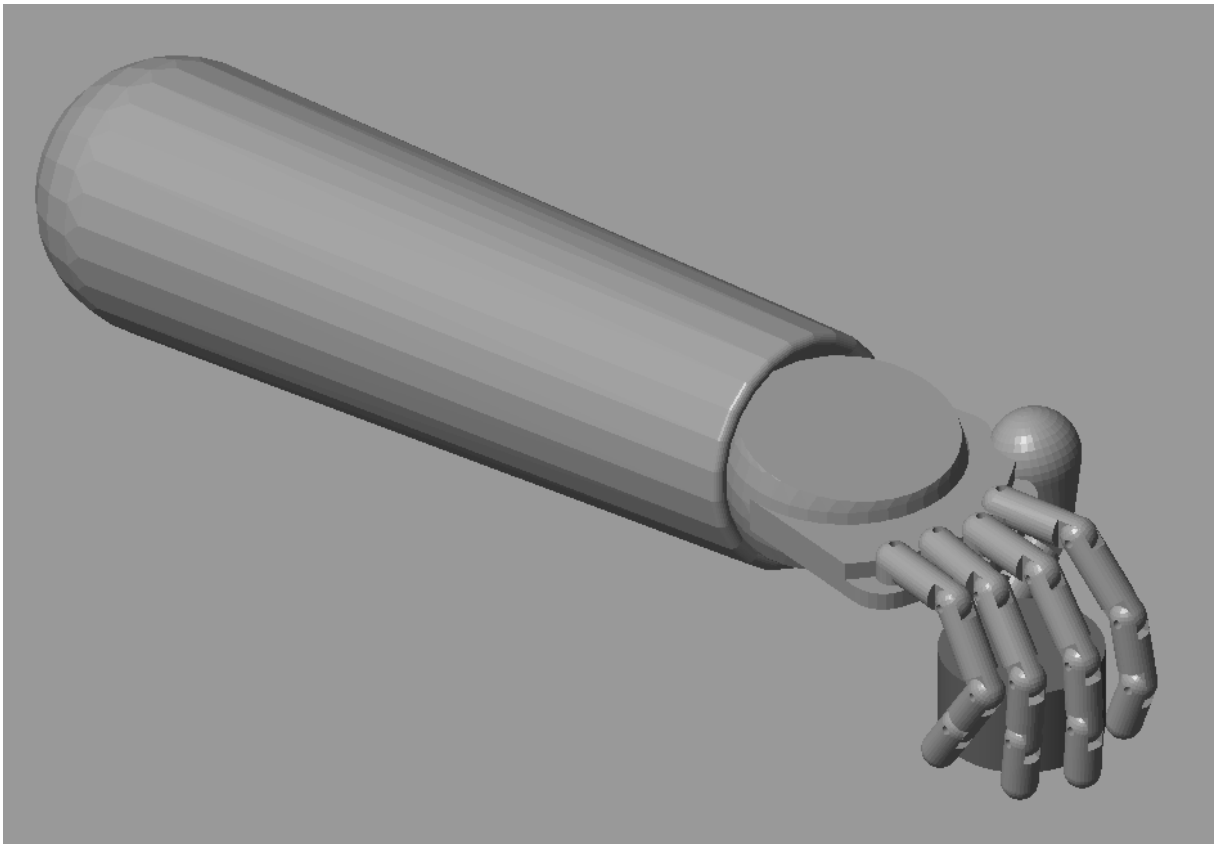


Obrázek 44: graf výpočtu fmincon

Výsledkem výpočtu inverzní kinematiky pomocí funkce fmincon je vektor natočení fi, ze kterého dopočítáme polohu a rozdíl polohy od žádané a výsledná hodnota cílové funkce 16,3066. Výpočet konvergoval po 501 iteracích (viz obr. 44). Výpočet trval 192,3 sekund.

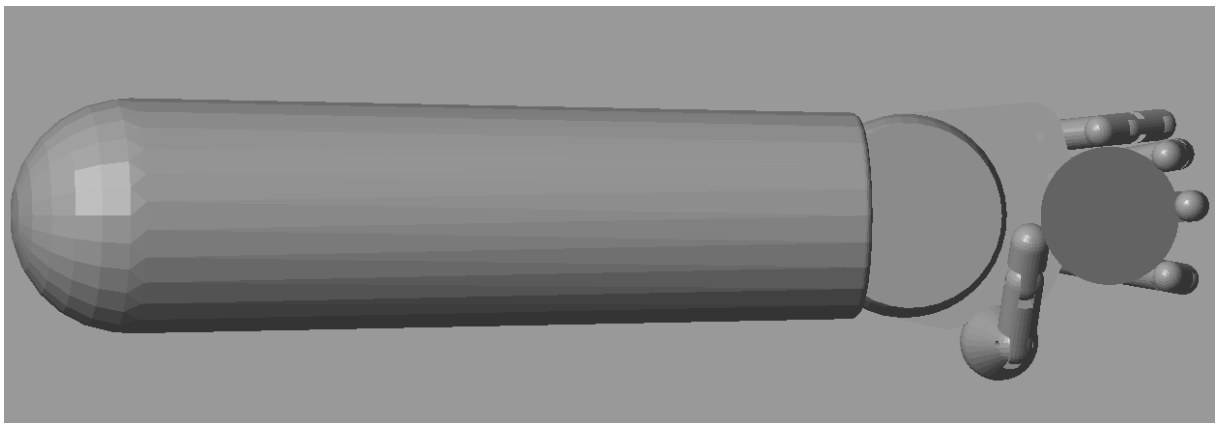
```
xpalec = [5.74999e+02;-4.14525e-04;2.1961e-04;1]
xukazovacek = [6.6035e+02;-35.35;2.195e-04;1]
xprostrednicek [6.74999e+02;-1.189e-04;-4.039e-05;1]
xprstenicek = [6.4999e+02;43.301;2.31e-04;1]
xmalicek = [5.99e+02;43.301;-7.936e-06;1]
```

```
vzdalenostPalec =   2.1086e-07
vzdalenostUkazovacek =   4.4560e-08
vzdalenostProstrednicek =   5.0730e-08
vzdalenostPrstenicek =   2.8436e-08
vzdalenostMalicek =   5.5353e-08
```



Obrázek 45:Ruka v úchopu1

Na obrázcích 45 a 46 je vidět úchop objektu podobně jako by to provedla lidská ruka. Na obrázku 45 je úchop v perspektivě a na obrázku 46 je pohled zespodu.



Obrázek 46: Ruka v úchopu 2

6 ZÁVĚR

Byla vytvořena realistická simulace pěti zadaných gest. Gesta jsou prováděna tak, že napodobují přirozený pohyb lidské ruky. Senzor by tak neměl mít problém s rozpoznáním daného gesta.

V druhé části práce byl rozebrán projekt z hlediska softwarového inženýrství. Navrhly datové toky, potřebné databáze a prvky systému. Byla vypočtena orientační cena projektu, která činí přibližně 2 800 000 Kč. Byl zjištěn odhad doby trvání projektu a jeho časový harmonogram. Celková doba trvání projektu je odhadnuta na 142,5 dne.

V závěrečné části práce byly rozebrány možnosti výpočtu inverzní kinematiky pomocí metod pro globální a lokální optimalizaci. Výpočet inverzní kinematiky je možné provést a lze přidat optimalizační podmínky tak, aby výsledná poloha ruky byla přirozená. Jako výhodnější v tomto případě bylo použití funkce fmincon, které respektuje maximální možné polohy kloubů, na rozdíl od funkce fminsearch. Výpočet je oproti výpočtu pomocí genetického algoritmu řádově rychlejší. Výsledné nalezené minimum cílové funkce je také výrazně menší než při použití funkce ga s lokální optimalizací pomocí následné aplikace funkce fminsearch.

V projektu lze dále pokračovat například s vytvořením grafického rozhraní, které by usnadnilo dlouhodobé testování. Případně přechodem od simulace k hardwarovému testování senzoru gest, kde by robot mohl být řízen simulačním modelem.

7 ZDROJE

[1] Průmyslové roboty a manipulátory, Doc. Ing. Vladimír ANDRLÍK, CSc. a kol., Ústav výrobních strojů a zařízení, 2012

[2] http://csse.usc.edu/tools/cocomoii.php

[3] https://www.technikaatrh.cz/pohony/motory-maxon-pro-aplikace-v-humanoidni-robotice-a-biomechanice

[4] http://strojirenstvi-maturita.blogspot.cz/2011/10/11-maticove-metody.html

[5] https://leporelo.info/extrem-funkce

[6] http://www.directindustry.com/prod/schunk/product-12463-902399.html

[7] https://schunk.com/us_en/homepage/the-new-schunk-pgn-plus-and-pgn-plus-electric-grippers/

[8] https://www.openbionics.com/shop/

[9] http://www.robotshop.com/en/mechax-robot-right-hand-upgraded.html

[10] http://www.robotshop.com/en/biogripp-robotic-hand-3d-print.html

[11] http://www.active-robots.com/ar10-humanoid-robotic-hand

[12] https://schunk.com/cz_en/gripping-systems/series/svh/

[13] http://new.abb.com/products/robotics/industrial-robots/irb-120

[14] https://www.universal-robots.com/cs/v%C3%BDrobky/robot-ur3/

[15] https://www.robots.com/kuka/kr-6-r700-fivve

[16] http://www.getautomata.com/

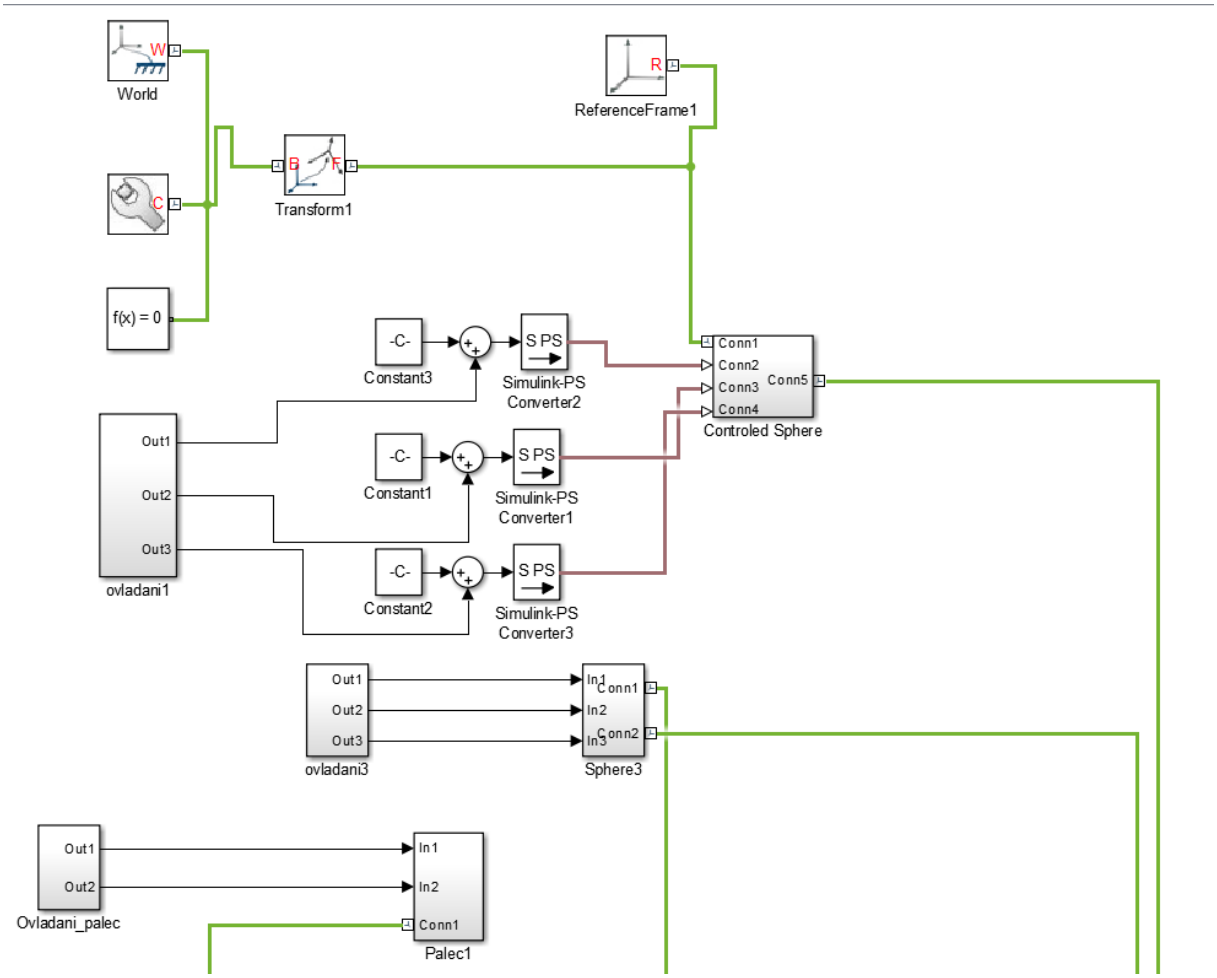
[17] BORTEL, Martin. *Evoluční algoritmy* [online]. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2012 [cit. 2017-06-28]. Dostupné z: http://hdl.handle.net/11012/9863. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací. Vedoucí práce Radim Burget.

8 PŘÍLOHY

8.1 MODEL V SIMULINKU

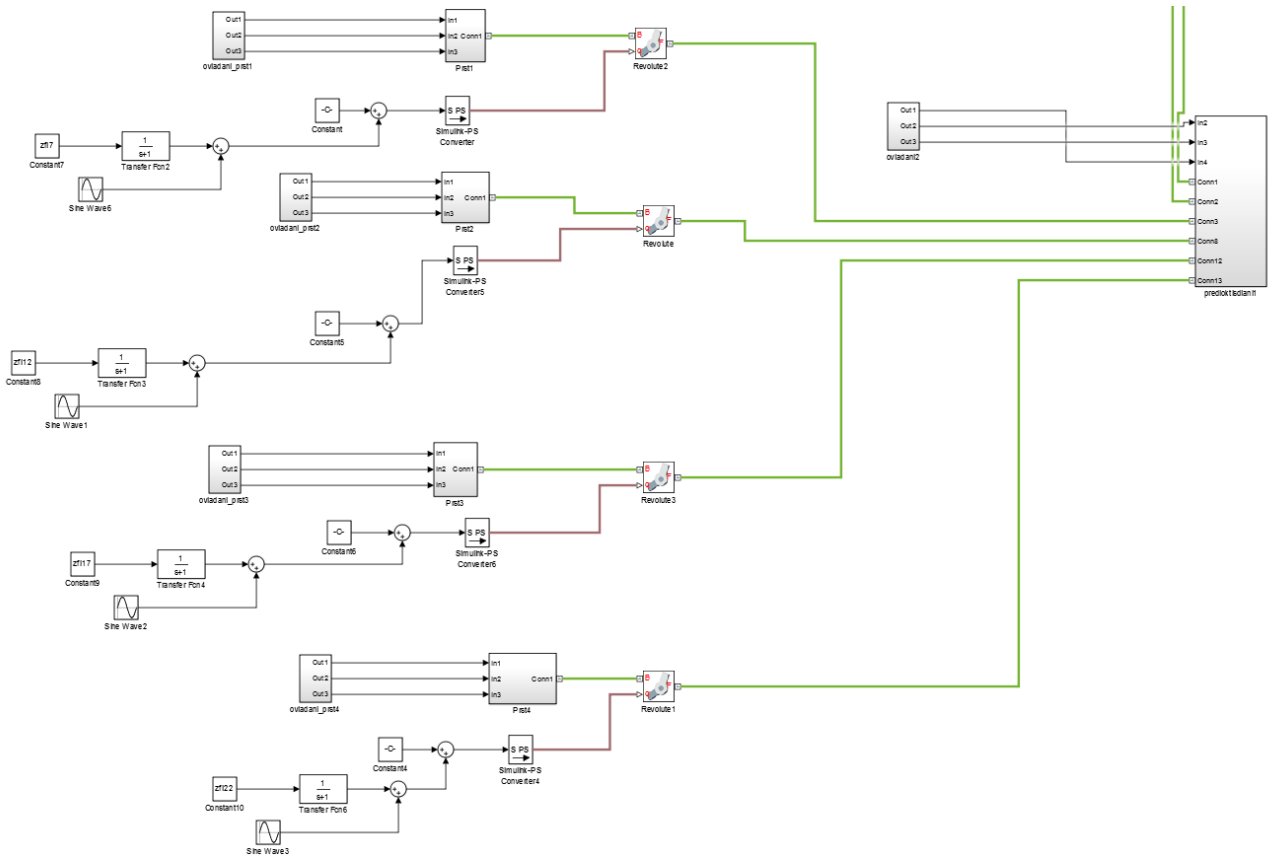
V této části příloh jsou výřezy ze simulinkového modelu ruky.

8.1.1 MODEL TOP



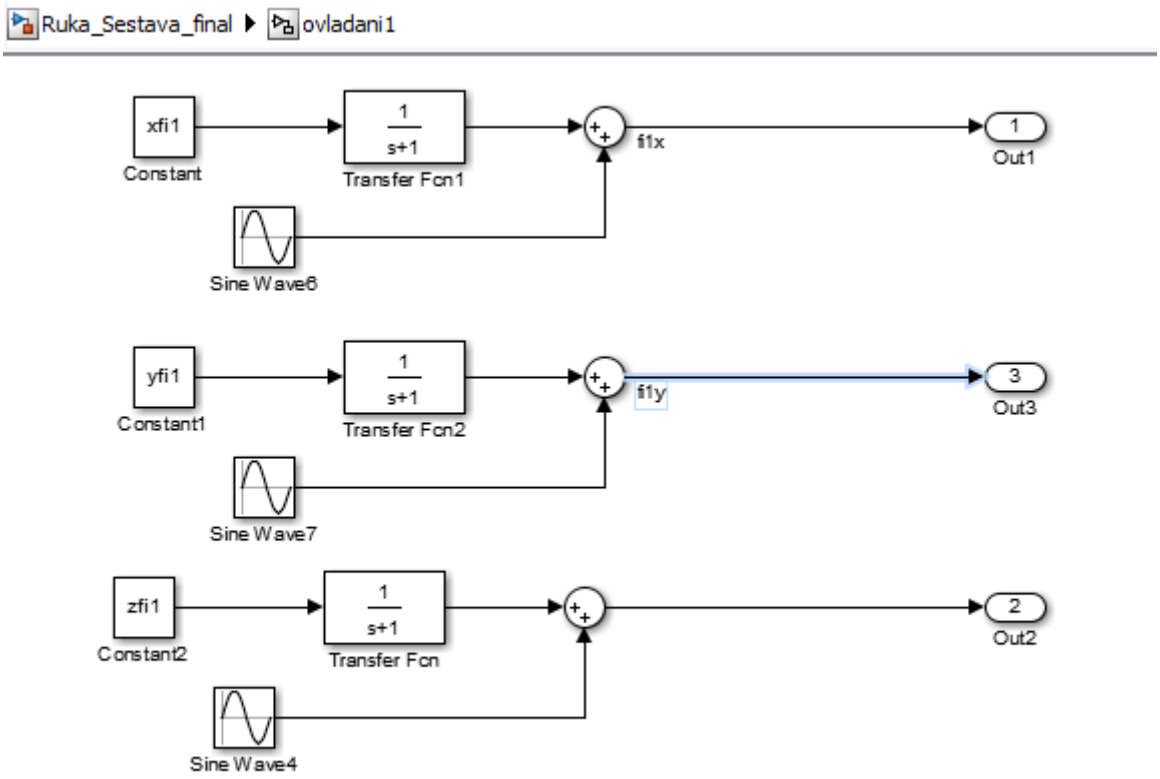
Obrázek 47: Vrchní část simulinkového modelu ruky

8.1.2 MODEL BOTTOM



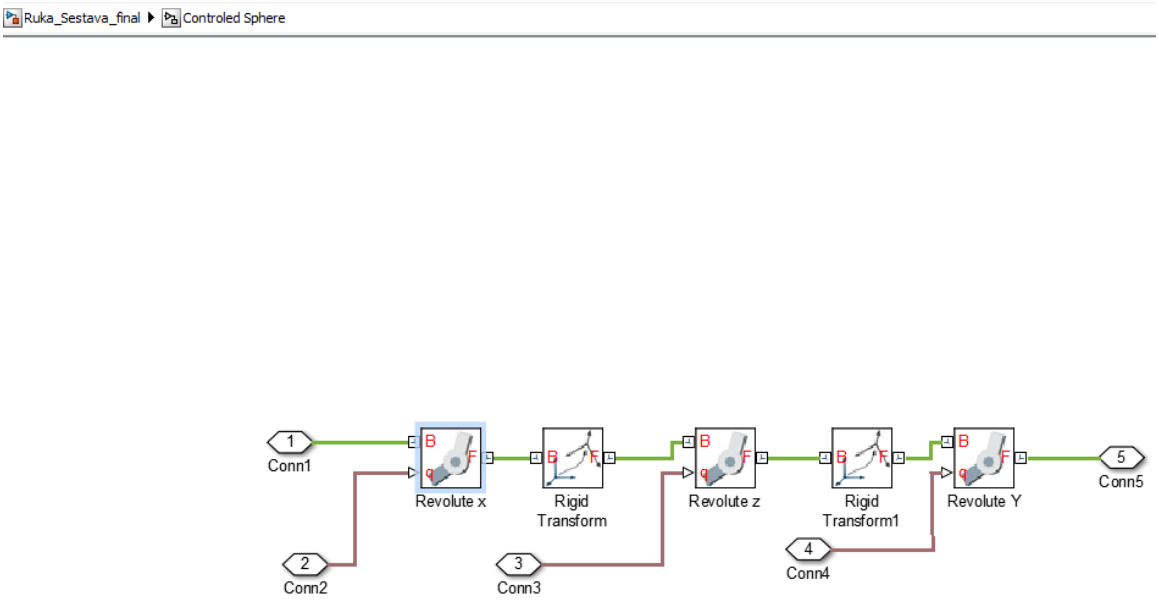
Obrázek 48: Spodní část simulinkového modelu

8.1.3 OVLADANI1



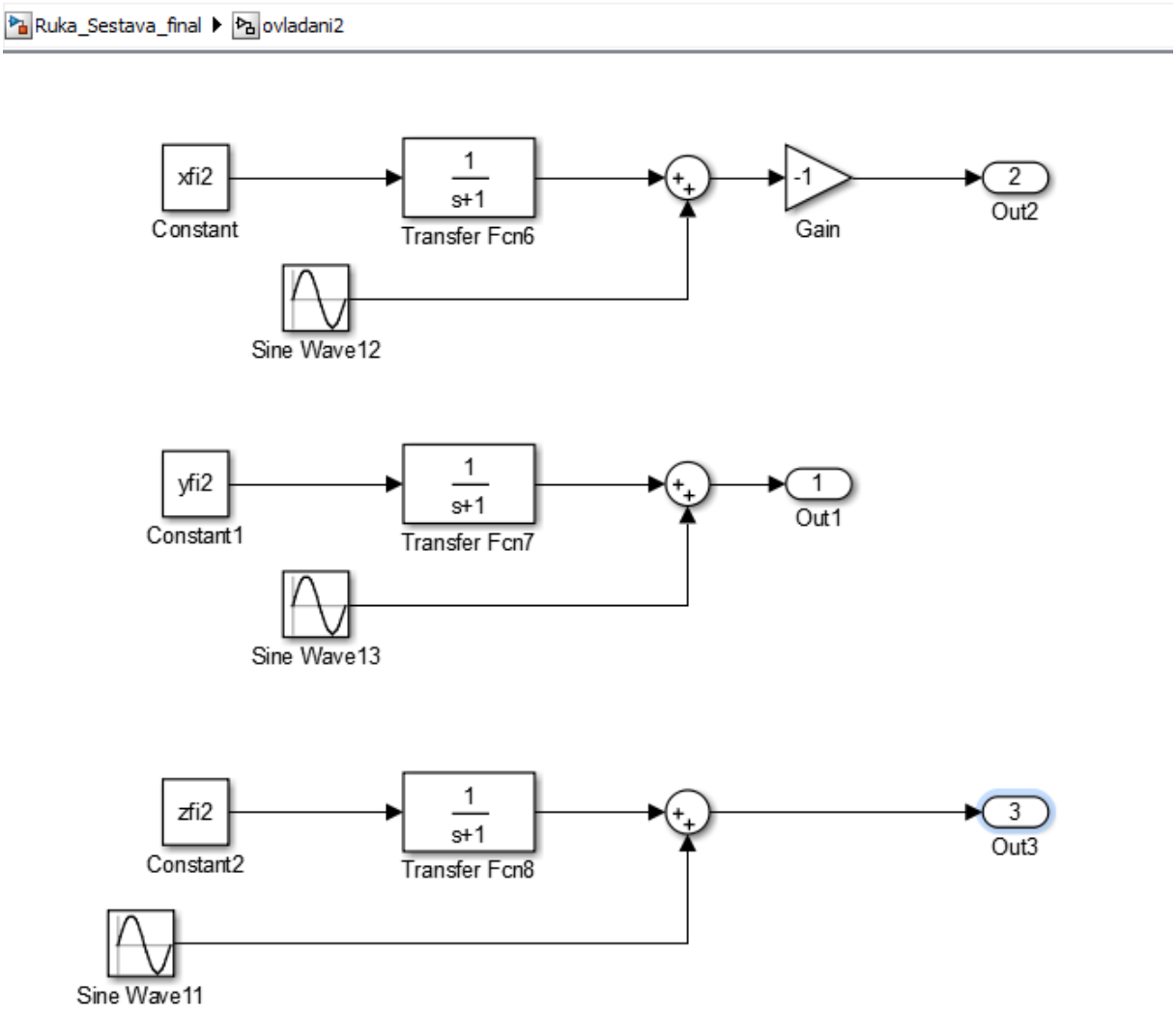
Obrázek 49: subsystém Ovladani1, blok ovládání první kulové vazby

8.1.4 SPHERE1



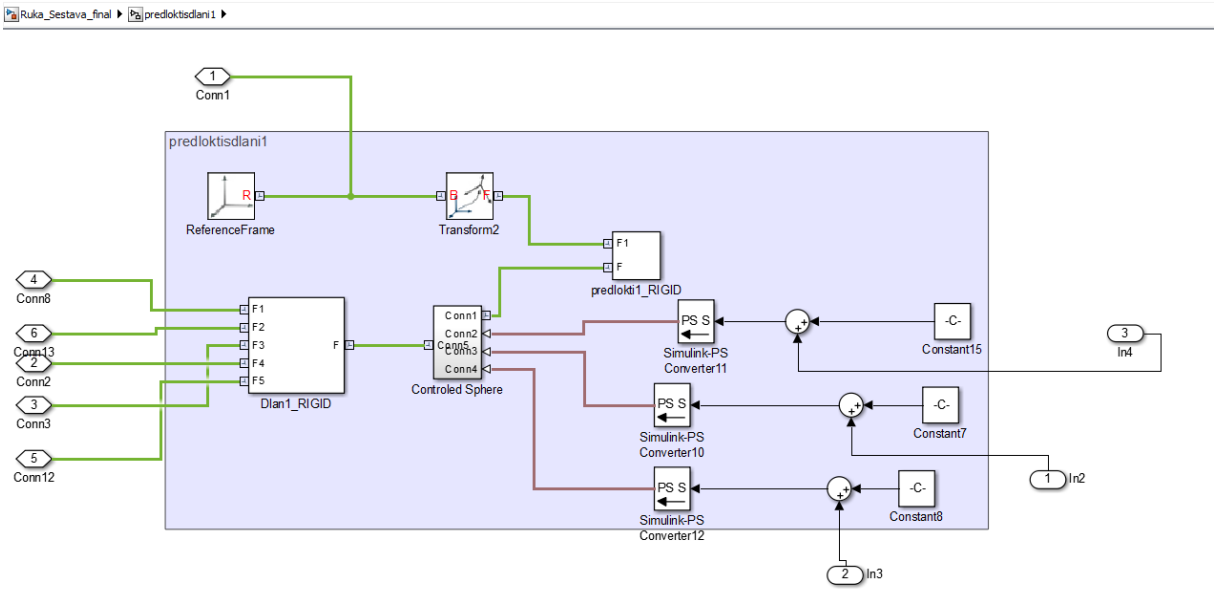
Obrázek 50: subsystém sphere1 blok první kulové vazby

8.1.5 OVLÁDÁNÍ2



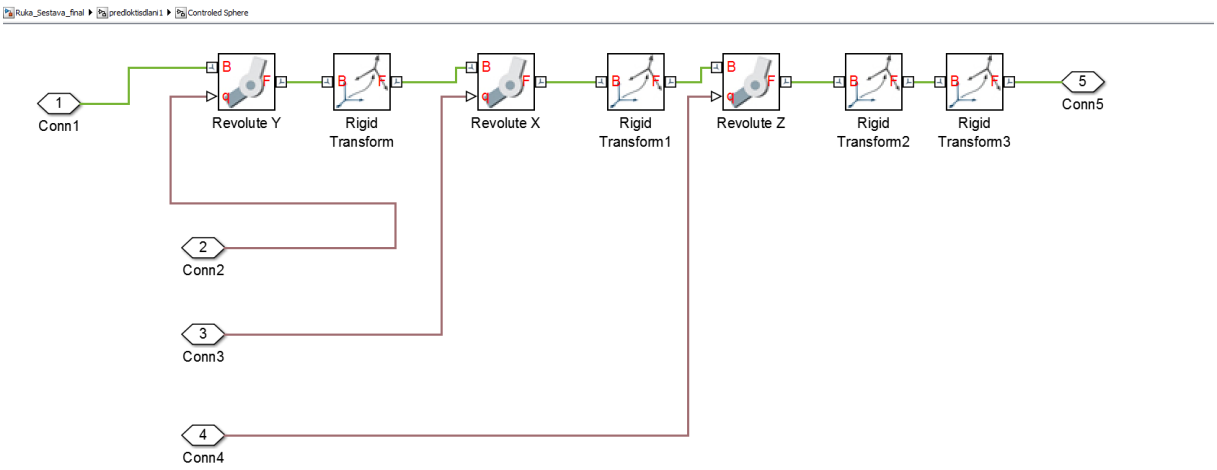
Obrázek 51: subsystem Ovladani2, blok ovládání druhé kulové vazby

8.1.6 PREDLOKTI S DLANI



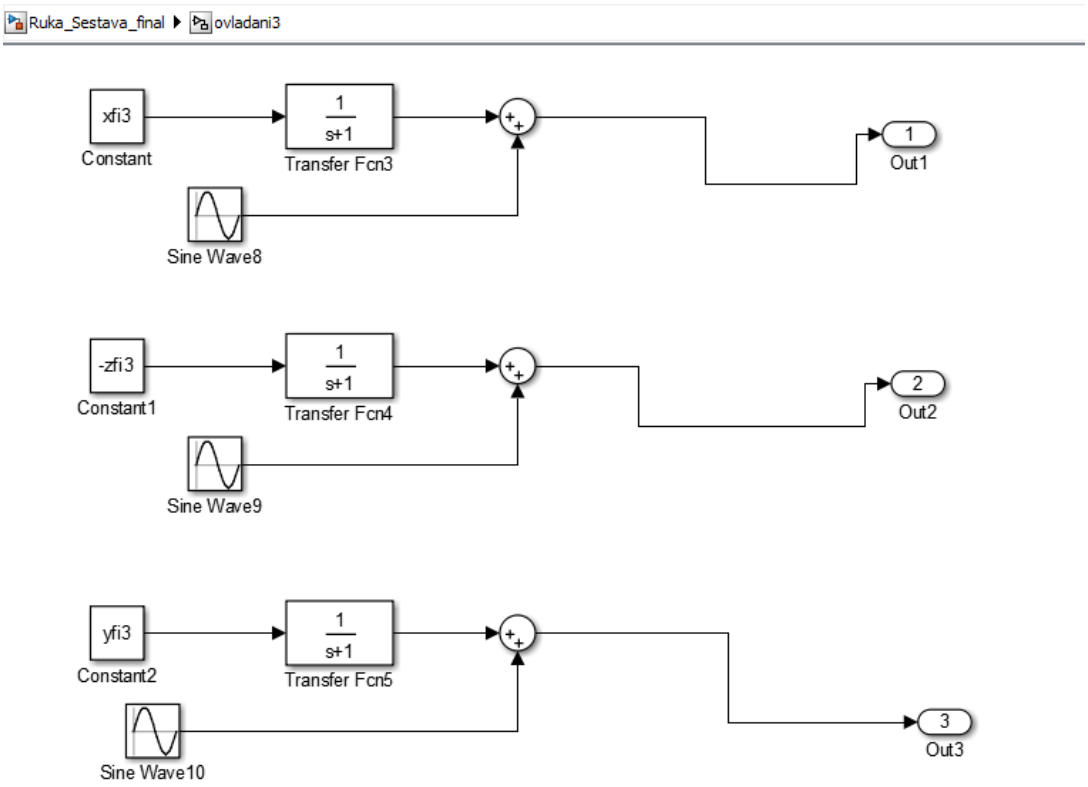
Obrázek 52: subsystém predloktisdlani

8.1.7 SPHERE 2



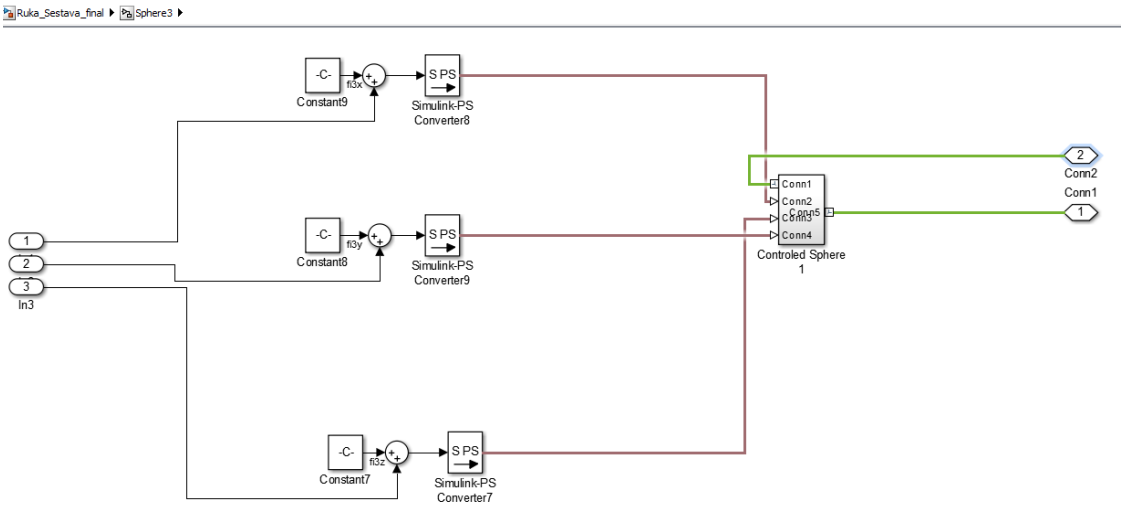
Obrázek 53: subsystem sphere2, blok druhé kulové vazby

8.1.8 OVLÁDÁNÍ3



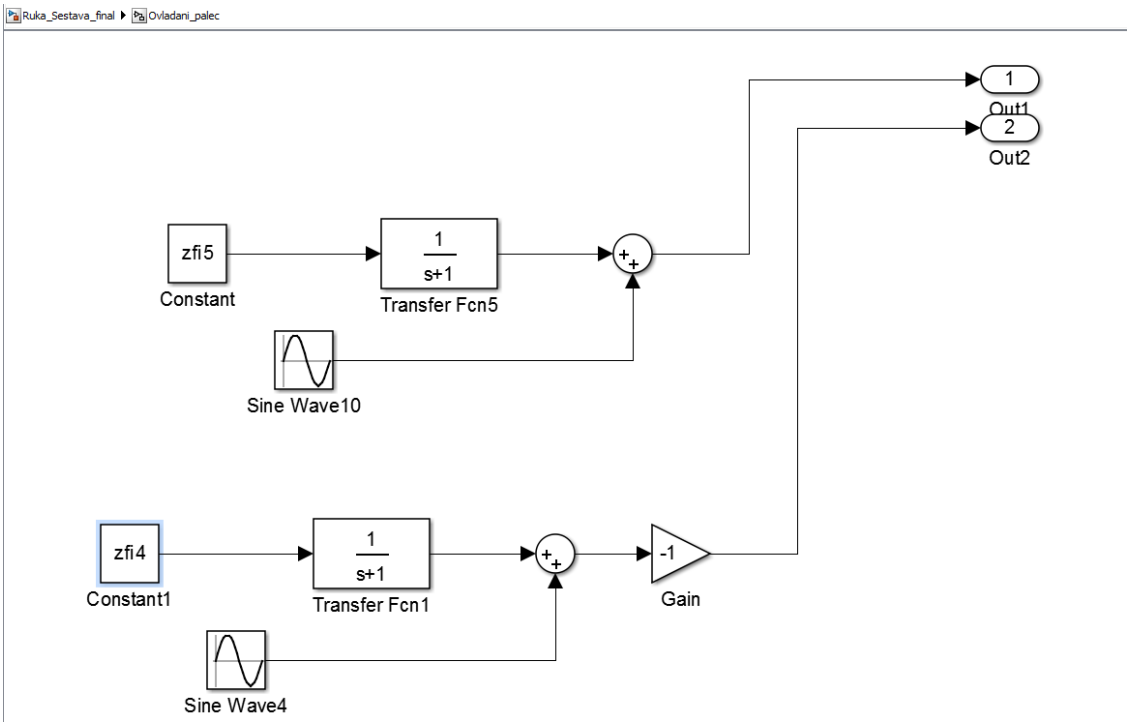
Obrázek 54: subsystém ovladani3, blok ovladaní třetí kulové vazby

8.1.9 SPHERE3



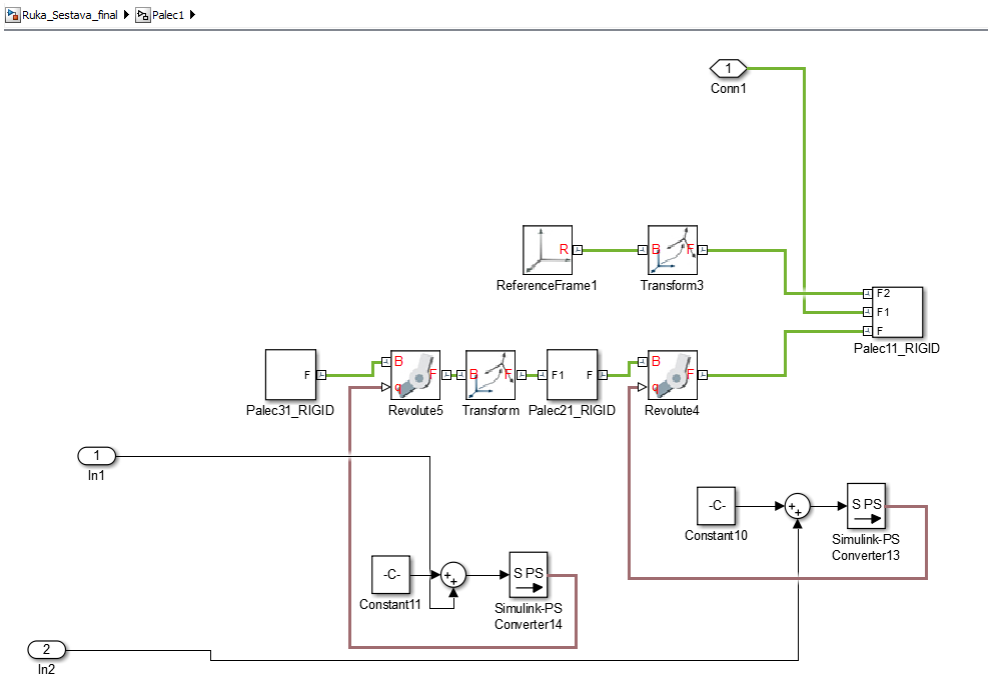
Obrázek 55: subsystém sphere3, blok 3 kulové vazby

8.1.10 OVLÁDÁNÍ PALEC



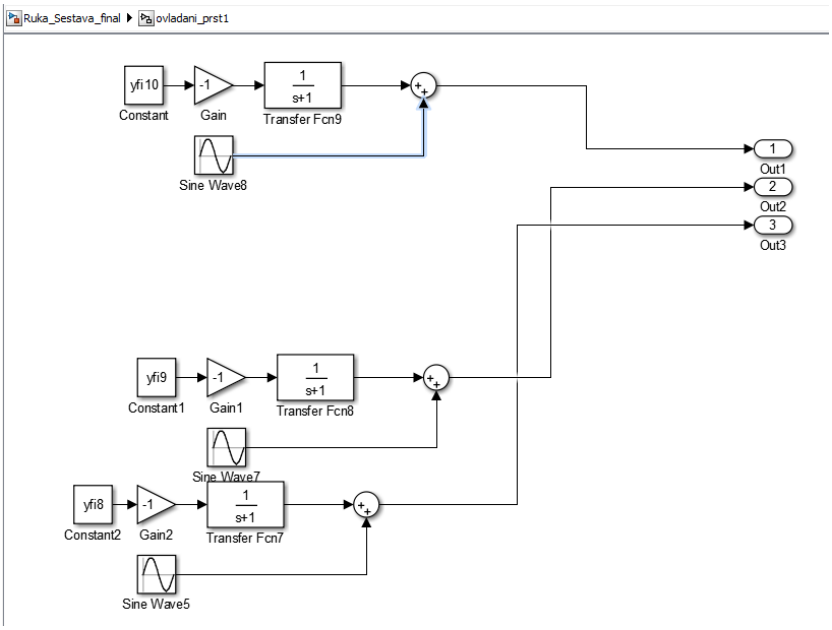
Obrázek 56: subsystém ovládající pohyb palce

8.1.11 PALEC



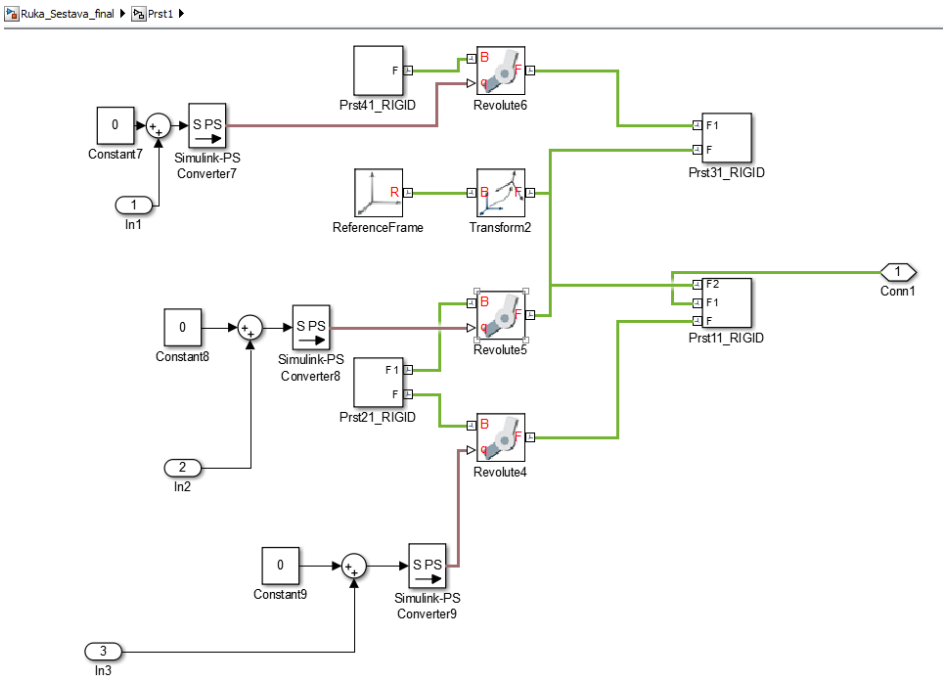
Obrázek 57: subsystém palce

8.1.12 OVLÁDÁNÍ PRST 1



8.1.13 PRST1

Obrázek 58: subsystém ovládání pohybu prstu



Obrázek 59: subsystém prstu

Ostatní prsty jsou dělané stejně.

8.2 MATLAB SKRIPTY

V další části jsou skripty Matlabu.

8.2.1 KONSTANTS

```
%konstanty
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222

a=500;
b=40;
c=50;
d=30;
e=50;
f=50;
g=40;
h=30;

i=e;
m=e;
q=e;
j=f;
n=f;
r=f;
k=g;
o=g;
s=g;
l=h;
p=h;
t=h;

x23=60;
y23=80;
x27=90;
y27=25;
x212=100;
y212=0;
x217=93;
y217=-25;
x222=86;
y222=-50;
```

8.2.2 MATICOVY_ZAPIS

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222

konstants;
 xfi1=fi(1);
 xfi2=fi(2);
```

```
xfi3=fi(3);
yfi1=fi(4);
yfi2=fi(5);
yfi3=fi(6);
yfi8=fi(7);
yfi9=fi(8);
yfi10=fi(9);
yfi13=fi(10);
yfi14=fi(11);
yfi15=fi(12);
yfi18=fi(13);
yfi19=fi(14);
yfi20=fi(15);
yfi23=fi(16);
yfi24=fi(17);
yfi25=fi(18);
zfi1=fi(19);
zfi2=fi(20);
zfi3=fi(21);
zfi4=fi(22);
zfi5=fi(23);
zfi7=fi(24);
zfi12=fi(25);
zfi17=fi(26);
zfi22=fi(27);



%% zapesti
T12=Txfi(xfi1)*Tyfi(yfi1)*Tzfi(zfi1)*Tx(a);

%% palec
T23=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x23)*Ty(y23);
T34=Tyfi(yfi3)*Tzfi(zfi3)*Txfi(xfi3)*Tx(b);
T45=Tzfi(zfi4)*Tx(c);
T56=Tzfi(zfi5)*Tx(d);

%% ukazovacek
T27=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x27)*Ty(y27);
T78=Tzfi(zfi7)*Tx(e);
T89=Tyfi(yfi8)*Tx(f);
T910=Tyfi(yfi9)*Tx(g);
T1011=Tyfi(yfi10)*Tx(h);

%% prostrednicek
T212=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x212)*Ty(y212);
T1213=Tzfi(zfi12)*Tx(i);
T1314=Tyfi(yfi13)*Tx(j);
T1415=Tyfi(yfi14)*Tx(k);
T1516=Tyfi(yfi15)*Tx(l);

%% prstenicek
T217=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x217)*Ty(y217);
T1718=Tzfi(zfi17)*Tx(m);
T1819=Tyfi(yfi18)*Tx(n);
T1920=Tyfi(yfi19)*Tx(o);
T2021=Tyfi(yfi20)*Tx(p);
```

```
%% malicek
T222=Txfi(xfi2)*Tyfi(yfi2)*Tzfi(zfi2)*Tx(x222)*Ty(y222);
T2223=Tzfi(zfi22)*Tx(q);
T2324=Tyfi(yfi23)*Tx(r);
T2425=Tyfi(yfi24)*Tx(s);
T2526=Tyfi(yfi25)*Tx(t);

%% transformace

Tpalec=T12*T23*T34*T45*T56;
Tukazovacek=T12*T27*T78*T89*T910*T1011;
Tprostrednicek=T12*T212*T1213*T1314*T1415*T1516;
Tprstenicek=T12*T217*T1718*T1819*T1920*T2021;
Tmalicek=T12*T222*T2223*T2324*T2425*T2526;

T=[Tpalec
   Tukazovacek
   Tprostrednicek
   Tprstenicek
   Tmalicek];

x0=[0
    0
    0
    1];

xpalec=Tpalec*x0;
xukazovacek=Tukazovacek*x0;
xprostrednicek=Tprostrednicek*x0;
xprstenicek=Tprstenicek*x0;
xmalicek=Tmalicek*x0;
xtop=T12*x0;
X4=T12*T23*T34*x0;
X = [xpalec
    xukazovacek
    xprostrednicek
    xprstenicek
    xmalicek];

%% kruh

stred=[625, 0, 0];

polomer=50;

palecSouradnice=[stred(1)+polomer*cos(pi)
                 stred(2)+polomer*sin(pi)
                 0];
ukazovacekSouradnice=[stred(1)+polomer*cos(-45*pi/180)
                      stred(2)+polomer*sin(-45*pi/180)
                      0];
prostrednicekSouradnice=[stred(1)+polomer*cos(0)
                         stred(2)+polomer*sin(0)
                         0];
prstenicekSouradnice=[stred(1)+polomer*cos(60*pi/180)
```

```
                      stred(2)+polomer*sin(60*pi/180)
                      0]';
malicekSouradnice=[stred(1)+polomer*cos(120*pi/180)
                   stred(2)+polomer*sin(120*pi/180)
                   0]';
```

8.2.3 MYFITNESS

```
%fitness fce

function y = myFitness(fi)
maticovy_zapis;



rlx=abs(fi(1)-0);
rl=rlx;

firovnom=(abs(fi(1))+abs(fi(2))+abs(fi(3))+abs(fi(4))+abs(fi(5))+abs(fi(6))
+abs(fi(7))+abs(fi(8))+abs(fi(9))+abs(fi(10))+abs(fi(11))+abs(fi(12))+abs(f
i(13))+abs(fi(14))+abs(fi(15))+abs(fi(16))+abs(fi(17))+abs(fi(18))+abs(fi(1
9))+abs(fi(20))+abs(fi(21))+abs(fi(22))+abs(fi(23))+abs(fi(24))+abs(fi(25))
+abs(fi(26))+abs(fi(27)));
% dlantopx=abs(xtop(1)-0)
% dlantopy=abs(xtop(2)-0)
dlantopz=abs(xtop(3)-100);
dlantop=dlantopz;


vzdalenostPalec=(((xpalec(1)-palecSouradnice(1))^2)+((xpalec(2)-
palecSouradnice(2))^2)+(xpalec(3)-palecSouradnice(3))^2)^1/2;
vzdalenostUkazovacek=(((xukazovacek(1)-
ukazovacekSouradnice(1))^2)+((xukazovacek(2)-
ukazovacekSouradnice(2))^2)+((xukazovacek(3)-
ukazovacekSouradnice(3))^2))^1/2;
vzdalenostProstrednicek=((xprostrednicek(1)-
prostrednicekSouradnice(1))^2+(xprostrednicek(2)-
prostrednicekSouradnice(2))^2+(xprostrednicek(3)-
prostrednicekSouradnice(3))^2)^1/2;
vzdalenostPrstenicek=((xprstenicek(1)-
prstenicekSouradnice(1))^2+(xprstenicek(2)-
prstenicekSouradnice(2))^2+(xprstenicek(3)-prstenicekSouradnice(3))^2)^1/2;
vzdalenostMalicek=((xmalicek(1)-malicekSouradnice(1))^2+(xmalicek(2)-
malicekSouradnice(2))^2+(xmalicek(3)-malicekSouradnice(3))^2)^1/2;

y = rl + 1*firovnom + 100*dlantop +
10000*(vzdalenostPalec+vzdalenostUkazovacek+vzdalenostProstrednicek+vzdalen
ostPrstenicek+vzdalenostMalicek);



end
```

8.2.4 GENETIC ALGORITHM + FMINSEARCH

```
% main function

tic

close all,
init;
LB;
UB;
fitfce=@myFitness;
nvars=27;
options = gaoptimset('Display','iter',... %display every iteration
'Generations',20000,... %maximum number of generations is 70
'TolFun',1e-7,... %tolerance for optimisation is 0.01
'TolCon',1e-7,...
'PlotFcns',@gaplotbestf);

[fi,fval,exitflag,output,population]=ga(fitfce, nvars, [],[],[],[],lb,
ub,[], options);

fi2=fi
% % nvars=27;
% % fitfce2=@myFitness2;
% % LB2;
% % UB2;
%
options = optimset('PlotFcns',@optimplotfval,...
    'TolFun',1e-7,... %tolerance for optimisation is 0.01
    'TolCon',1e-7,...
    'MaxFunEvals',300000,...
    'MaxIter',100000);
[x,fval2,exitflag2,output2]=fminsearch(@myFitness,fi, options);
% [fi,fval,exitflag,output,population]=ga(fitfce2, nvars, [],[],[],[],lb2,
ub2,[], options);

maticovy_zapis;
fi

toc
```

8.2.5 LB

```
% % spodni hranice
%
 lb=[ -pi/2  % xfi1=fi(1);
-pi/2%  xfi2=fi(2);
-pi/2%  xfi3=fi(3);
-pi/2%  yfi1=fi(4);
-pi/2%  yfi2=fi(5);
```

```
-pi/2%  yfi3=fi(6);
0%  yfi8=fi(7);
0%  yfi9=fi(8);
0%  yfi10=fi(9);
0%  yfi13=fi(10);
0%  yfi14=fi(11);
0%  yfi15=fi(12);
0%  yfi18=fi(13);
0%  yfi19=fi(14);
0%  yfi20=fi(15);
0
0%  yfi24=fi(17);
0%  yfi25=fi(18);
-pi/2%  zfi1=fi(19);
-pi/2%  zfi2=fi(20);
-pi/2%  zfi3=fi(21);
0%  zfi4=fi(22);
0%  zfi5=fi(23);
-10*pi/180%  zfi7=fi(24);
-10*pi/180%  zfi12=fi(25);
-10*pi/180%  zfi17=fi(26);
-10*pi/180]%  zfi22=fi(27);
```

8.2.6 UB

```
% % horni hranice
%
ub=[ pi/2  % xfi1=fi(1);
pi/2%  xfi2=fi(2);
0%  xfi3=fi(3);
pi/2%  yfi1=fi(4);
pi/2%  yfi2=fi(5);
pi/2%  yfi3=fi(6);
pi/2%  yfi8=fi(7);
pi/2%  yfi9=fi(8);
pi/2%  yfi10=fi(9);
pi/2%  yfi13=fi(10);
pi/2%  yfi14=fi(11);
pi/2%  yfi15=fi(12);
pi/2%  yfi18=fi(13);
pi/2%  yfi19=fi(14);
pi/2%  yfi20=fi(15);
pi/2
pi/2%  yfi24=fi(17);
pi/2%  yfi25=fi(18);
pi/2%  zfi1=fi(19);
pi/2%  zfi2=fi(20);
0%  zfi3=fi(21);
pi/2%  zfi4=fi(22);
pi/2%  zfi5=fi(23);
10*pi/180%  zfi7=fi(24);
10*pi/180%  zfi12=fi(25);
10*pi/180%  zfi17=fi(26);
10*pi/180]%  zfi22=fi(27);
```

8.2.7 FMINCON

```
% main function

tic

fi=zeros(27,1);
LB;
UB;
fitfce=@myFitness;
options
=optimset('display','iter','MaxIter',100000,'MaxFunEval',5000000,'TolX',1e-
8,'TolFun',1e-8,'PlotFcns',@optimplotfval);
[fi,fval,exitflag,output,lambda,grad,hessian] =
fmincon(fitfce,fi,[],[],[],[],lb,ub,[],options);

maticovy_zapis;

vzdalenostPalec=(((xpalec(1)-palecSouradnice(1))^2)+((xpalec(2)-
palecSouradnice(2))^2)+(xpalec(3)-palecSouradnice(3))^2)^1/2
vzdalenostUkazovacek=(((xukazovacek(1)-
ukazovacekSouradnice(1))^2)+((xukazovacek(2)-
ukazovacekSouradnice(2))^2)+((xukazovacek(3)-
ukazovacekSouradnice(3))^2))^1/2
vzdalenostProstrednicek=((xprostrednicek(1)-
prostrednicekSouradnice(1))^2+(xprostrednicek(2)-
prostrednicekSouradnice(2))^2+(xprostrednicek(3)-
prostrednicekSouradnice(3))^2)^1/2
vzdalenostPrstenicek=((xprstenicek(1)-
prstenicekSouradnice(1))^2+(xprstenicek(2)-
prstenicekSouradnice(2))^2+(xprstenicek(3)-prstenicekSouradnice(3))^2)^1/2
vzdalenostMalicek=((xmalicek(1)-malicekSouradnice(1))^2+(xmalicek(2)-
malicekSouradnice(2))^2+(xmalicek(3)-malicekSouradnice(3))^2)^1/2

xpalec

toc
```

8.2.8 INIT

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
```

```
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m


%% statika

xfi1=0;
yfi1=0;
zfi1=0;

xfi2=0;
yfi2=0;
zfi2=0;

% palec
xfi3=0;
yfi3=0;
zfi3=0;

zfi4=0;
zfi5=0;

% ukazovacek
zfi7=0;
yfi8=0;
yfi9=0;
yfi10=0;

% prostrednicek
zfi12=0;
yfi13=0;
yfi14=0;
yfi15=0;

% prstenicek
zfi17=0;
yfi18=0;
yfi19=0;
yfi20=0;

% malicek
zfi22=0;
yfi23=0;
yfi24=0;
yfi25=0;


%% pohyb

xfi1m=0;
yfi1m=0;
zfi1m=0;

xfi2m=0;
```

```
yfi2m=0;
zfi2m=0;

% palec
xfi3m=0;
yfi3m=0;
zfi3m=0;

zfi4m=0;
zfi5m=0;

% ukazovacek
zfi7m=0;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=0;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=0;
yfi18m=0;
yfi19m=0;
yfi20m=0;

% malicek
zfi22m=0;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```

8.2.9 GRIP

```
%% grip
init
 fi=[0%1.2950 % xfi1=fi(1);
    0%     0.7911  %xfi2=fi(2);
   -0.0690  %xfi3=fi(3);
    0.148%-0.0148  %yfi1=fi(4);
   -0.148%    -1.1442  %  yfi2=fi(5);
    -1.5203%     0.3285%  yfi3=fi(6);
    0.8934%     0.0004%  yfi8=fi(7);
    0.5%     1.2674%  yfi9=fi(8);
    0.5%     1.5701%  yfi10=fi(9);
    0.8934%  yfi13=fi(10);
    0.6111%  yfi14=fi(11);
    0.1379%  yfi15=fi(12);
    0.8625%  yfi18=fi(13);
    0.8%     0.9818%  yfi19=fi(14)
    0.0317%  yfi20=fi(15);
    1%     0.7721%  yfi23=fi(16);
```

```
    1.1553%  yfi24=fi(17);
    0%     1.5203%  yfi25=fi(18);
    0%     0.2052%  zfi1=fi(19);
    0   % -0.6417%  zfi2=fi(20)
    -0.3%    -1.0495%  zfi3=fi(21);
   -1%     0.2249 %zfi4=fi(22);
    0.0163%  zfi5=fi(23);
   -0.1744%     0.1744%  zfi7=fi(24);
    0.0751%  zfi12=fi(25);
    0.1743%    -0.1743%  zfi17=fi(26);
    0.1042];%  zfi22=fi(27);
maticovy_zapis;
```

8.2.10 CHANGE SONG

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m


%% statika

xfi1=-90*pi/180;
yfi1=0;
zfi1=0;

xfi2=0;
yfi2=0;
zfi2=0;

% palec
xfi3=0;
yfi3=0;
zfi3=-90*pi/180;

zfi4=0;
zfi5=0;

% ukazovacek
zfi7=0;
yfi8=0;
yfi9=0;
yfi10=0;
```

```
% prostrednicek
zfi12=0;
yfi13=0;
yfi14=0;
yfi15=0;

% prstenicek
zfi17=0;
yfi18=0;
yfi19=0;
yfi20=0;

% malicek
zfi22=0;
yfi23=0;
yfi24=0;
yfi25=0;


%% pohyb

xfi1m=0;
yfi1m=20*pi/180;
zfi1m=0;

xfi2m=0;
yfi2m=0;
zfi2m=0;

% palec
xfi3m=0;
yfi3m=0;
zfi3m=0;

zfi4m=0;
zfi5m=0;

% ukazovacek
zfi7m=0;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=0;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=0;
yfi18m=0;
yfi19m=0;
yfi20m=0;
```

```
% malicek
zfi22m=0;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```

8.2.11 ONEFINGERPOINT

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m


%% statika

xfi1=0;
yfi1=0;
zfi1=5*pi/180;;

xfi2=0;
yfi2=0;
zfi2=5*pi/180;;

% palec
xfi3=0;
yfi3=-90*pi/180;
zfi3=0;

zfi4=90*pi/180;;
zfi5=45*pi/180;;

% ukazovacek
zfi7=-5*pi/180;;
yfi8=0;
yfi9=0;
yfi10=0;

% prostrednicek
zfi12=0;
yfi13=90*pi/180;
yfi14=90*pi/180;
yfi15=40*pi/180;
```

```
% prstenicek
zfi17=0;
yfi18=90*pi/180;
yfi19=90*pi/180;
yfi20=40*pi/180;

% malicek
zfi22=0;
yfi23=90*pi/180;
yfi24=90*pi/180;
yfi25=40*pi/180;


%% pohyb

xfi1m=0;
yfi1m=0;
zfi1m=10*pi/180;

xfi2m=0;
yfi2m=0;
zfi2m=10*pi/180;;

% palec
xfi3m=0;
yfi3m=0;
zfi3m=0;

zfi4m=0;
zfi5m=0;

% ukazovacek
zfi7m=10*pi/180;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=0;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=0;
yfi18m=0;
yfi19m=0;
yfi20m=0;

% malicek
zfi22m=0;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```

8.2.12 TWOFINGERPOINT

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m


%% statika

xfi1=0;
yfi1=0;
zfi1=5*pi/180;

xfi2=0;
yfi2=0;
zfi2=5*pi/180;

% palec
xfi3=0;
yfi3=-90*pi/180;
zfi3=0;

zfi4=-90*pi/180;
zfi5=-90*pi/180;

% ukazovacek
zfi7=0;
yfi8=0;
yfi9=0;
yfi10=0;

% prostrednicek
zfi12=0;
yfi13=0;
yfi14=0;
yfi15=0;

% prstenicek
zfi17=0;
yfi18=90*pi/180;
yfi19=90*pi/180;
yfi20=40*pi/180;

% malicek
zfi22=0;
yfi23=90*pi/180;
```

```
yfi24=90*pi/180;
yfi25=40*pi/180;


%% pohyb

xfi1m=0;
yfi1m=0;
zfi1m=-10*pi/180;

xfi2m=0;
yfi2m=0;
zfi2m=-10*pi/180;;

% palec
xfi3m=0;
yfi3m=0;
zfi3m=0;

zfi4m=0;
zfi5m=0;

% ukazovacek
zfi7m=0;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=0;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=0;
yfi18m=0;
yfi19m=0;
yfi20m=0;

% malicek
zfi22m=0;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```

8.2.13 START

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
```

```
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m


%% statika

xfi1=0;
yfi1=0;
zfi1=0;

xfi2=0;
yfi2=0;
zfi2=0;

% palec
xfi3=0;
yfi3=0;
zfi3=-45*pi/180;

zfi4=0;
zfi5=0;

% ukazovacek
zfi7=-10*pi/180;
yfi8=0;
yfi9=0;
yfi10=0;

% prostrednicek
zfi12=-5*pi/180;
yfi13=0;
yfi14=0;
yfi15=0;

% prstenicek
zfi17=5*pi/180;
yfi18=0;
yfi19=0;
yfi20=0;

% malicek
zfi22=10*pi/180;
yfi23=0;
yfi24=0;
yfi25=0;


%% pohyb

xfi1m=0;
yfi1m=0;
```

```
zfi1m=0;

xfi2m=0;
yfi2m=0;
zfi2m=0;

% palec
xfi3m=0;
yfi3m=0;
zfi3m=-45*pi/180;

zfi4m=0;
zfi5m=0;

% ukazovacek
zfi7m=-10*pi/180;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=-5*pi/180;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=5*pi/180;
yfi18m=0;
yfi19m=0;
yfi20m=0;

% malicek
zfi22m=10*pi/180;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```

8.2.14 WAVE

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m
```

```
%% statika

xfi1=0;
yfi1=0;;
zfi1=0;

xfi2=0;
yfi2=0;;
zfi2=0;

% palec
xfi3=0;
yfi3=0;
zfi3=-90*pi/180;

zfi4=0;
zfi5=0;

% ukazovacek
zfi7=0;
yfi8=0;
yfi9=0;
yfi10=0;

% prostrednicek
zfi12=0;
yfi13=0;
yfi14=0;
yfi15=0;

% prstenicek
zfi17=0;
yfi18=0;
yfi19=0;
yfi20=0;

% malicek
zfi22=0;
yfi23=0;
yfi24=0;
yfi25=0;


%% pohyb

xfi1m=0;
yfi1m=10*pi/180;
zfi1m=0;

xfi2m=0;
yfi2m=20*pi/180;
zfi2m=0;
```

```
% palec
xfi3m=0;
yfi3m=0;
zfi3m=0;

zfi4m=0;
zfi5m=0;

% ukazovacek
zfi7m=0;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=0;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=0;
yfi18m=0;
yfi19m=0;
yfi20m=0;

% malicek
zfi22m=0;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```

8.2.15 THUMB UP

```
global a b c d e f g h i j k l m n o p q r s t x23 x27 x212 x217 x222 y23
y27 y212 y217 y222
global xfi1 xfi2 xfi3
global yfi1 yfi2 yfi3 yfi8 yfi9 yfi10 yfi13 yfi14 yfi15 yfi18 yfi19 yfi20
yfi23 yfi24 yfi25
global zfi1 zfi2 zfi3 zfi4 zfi5 zfi7 zfi12 zfi17 zfi22

global xfi1m xfi2m xfi3m
global yfi1m yfi2m yfi3m yfi8m yfi9m yfi10m yfi13m yfi14m yfi15m yfi18m
yfi19m yfi20m yfi23m yfi24m yfi25m
global zfi1m zfi2m zfi3m zfi4m zfi5m zfi7m zfi12m zfi17m zfi22m


%% statika

xfi1=-90*pi/180;
yfi1=0;
zfi1=0;
```

```
xfi2=0;
yfi2=0;
zfi2=0;

% palec
xfi3=0;
yfi3=0;
zfi3=0;

zfi4=0;
zfi5=0;

% ukazovacek
zfi7=0;
yfi8=90*pi/180;
yfi9=90*pi/180;
yfi10=40*pi/180;

% prostrednicek
zfi12=0;
yfi13=90*pi/180;
yfi14=90*pi/180;
yfi15=40*pi/180;

% prstenicek
zfi17=0;
yfi18=90*pi/180;
yfi19=90*pi/180;
yfi20=40*pi/180;

% malicek
zfi22=0;
yfi23=90*pi/180;
yfi24=90*pi/180;
yfi25=40*pi/180;



%% pohyb

xfi1m=0;
yfi1m=0;
zfi1m=0;

xfi2m=0;
yfi2m=0;
zfi2m=0;

% palec
xfi3m=0;
yfi3m=0;
zfi3m=0;

zfi4m=0;
```

```
zfi5m=0;

% ukazovacek
zfi7m=0;
yfi8m=0;
yfi9m=0;
yfi10m=0;

% prostrednicek
zfi12m=0;
yfi13m=0;
yfi14m=0;
yfi15m=0;

% prstenicek
zfi17m=0;
yfi18m=0;
yfi19m=0;
yfi20m=0;

% malicek
zfi22m=0;
yfi23m=0;
yfi24m=0;
yfi25m=0;
```